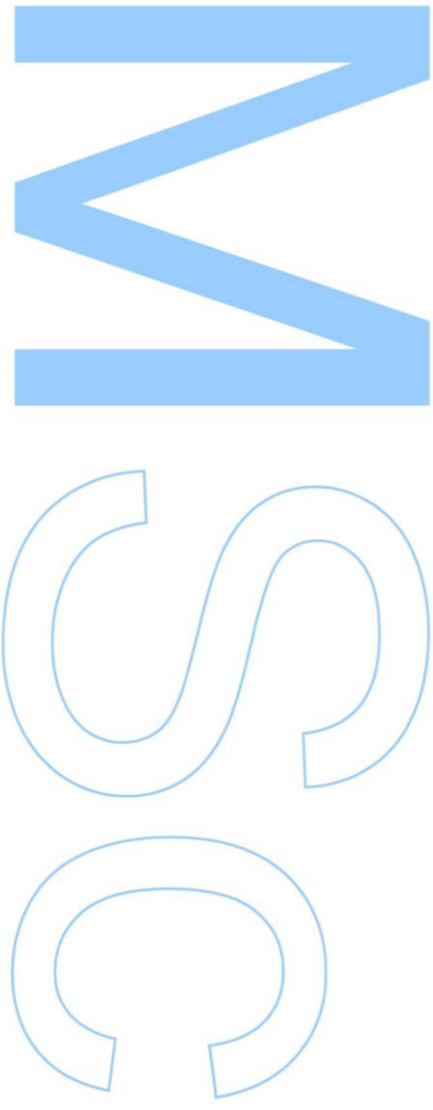
Domain Oriented Biclustering Validation

Carlos Alberto Magalhães Leite
Mestrado em Ciência de Computadores
Departamento de Ciência de Computadores
2016

Orientador
Luís Fernando Rainho Alves Torgo,
Professor Associado,
Faculdade de Ciências da Universidade do Porto

Coorientador
Catarina Maria Pinto Mora Pinto de Magalhães,
Professora Auxiliar Convidada,
Faculdade de Ciências da Universidade do Porto



Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, _____/_____/_____

To my parents and grandparents

FCT Fundação para a Ciência e a Tecnologia

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

Acknowledgments

I would like to thank to my supervisor, Professor Luís Torgo, for all the support, suggestions and corrections that greatly improved this dissertation. I am truly grateful for the opportunity to work in this project. Also, I want to thank to my co-supervisor, Catarina Magalhães, for the time, guidance, kindness and support that she demonstrated along the way. Without her positive attitude this work could not be accomplished.

A special mention to Professor Vítor Santos Costa for the insights about our case study, and to my friend Vítor Cerqueira for the helpful discussions about data mining.

Finally, I want to express my deep gratitude to my parents, my sister and my friends. A special thanks to my grandparents for everything that they thought me.

Last, but not least, I would like to thank to you, Rafaela. Without your infinite patience and kindness I would not be where I am today. I am very lucky to have met such an amazing person.

This work was financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project POCI-01-0145-FEDER-006961, and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013.

This work was also partial supported by EEA Grants Iceland, Lichtenstein, Norway, through MarinEye project, PT02_Aviso4_0017.

Carlos Leite

Porto, 2016

Resumo

Clustering é uma das tarefas mais utilizadas na área de Análise de Dados para particionar um conjunto de objetos. O particionamento dos objetos é realizado de forma a que os objetos pertencentes ao mesmo subconjunto (*cluster*) sejam similares entre si e dissimilares em relação aos objetos pertencentes a outros subconjuntos. Além disso, os objetos pertencentes ao mesmo cluster são similares entre si tendo em conta todos os atributos que os descrevem. Contudo, existem outras formulações do problema de *clustering*, para além da apresentada. Por exemplo, o objetivo pode ser encontrar grupos de objetos com um padrão semelhante em certos atributos, e não em todos eles. As técnicas de *biclustering* agrupam simultaneamente as linhas e as colunas de forma a encontrar esse tipo de grupos.

A motivação para o trabalho apresentado nesta dissertação surge através da aplicação de técnicas de *biclustering* ao conjunto de dados da metagenómica provenientes da iniciativa *Ocean Sampling Day*, realizada em 2014 à escala mundial. Uma vez que a atividade microbiana é uma componente fundamental dos ciclos biogeoquímicos dos oceanos, nós tentamos encontrar nichos geográficos de certas funções microbianas através da aplicação de técnicas de *biclustering*. O problema reside então em como determinar a relevância de um *bicluster* de um ponto de vista biológico e geográfico.

Nós propomos então uma metodologia geral que avalia um *bicluster* tendo em consideração a relevância das suas linhas e das suas colunas. Tais relevâncias são calculadas tendo em conta um conjunto de indicadores definidos de acordo com o domínio da aplicação. No nosso caso de estudo, a relevância das linhas corresponde à relevância

biológica, uma vez que as linhas nos dados da metagenómica representam as funções microbianas. Por outro lado, a relevância das colunas corresponde à relevância geográfica, uma vez que as colunas nos dados da metagenómica representam locais de amostragem.

Por conseguinte, aplicamos a metodologia proposta ao nosso caso de estudo utilizando uma aplicação, `ORCA`, que desenvolvemos no âmbito desta dissertação. A metodologia que propusemos permitiu-nos encontrar *biclusters* relevantes de um ponto de vista biológico e geográfico. Para além disso, também nos permitiu encontrar relações interessantes, que eram desconhecidas até à data, entre funções microbianas chave (transformações do ciclo do azoto) entre diferentes ecossistemas marinhos. Muitas dessas interrelações funcionais identificadas com a nossa metodologia são relevantes de um ponto de vista biológico.

Palavras-chave: clustering, biclustering, validação de biclusters, dados metagenómicos.

Abstract

Clustering is a traditional data mining task which consists in partitioning a set of data objects into subsets in such a way that the objects within the same subset (cluster) are similar to one another and dissimilar to objects in other subsets. Moreover, the objects in the same cluster are similar with respect to all the attributes (or features) that describe them. However, there are other formulations of the clustering problem. For instance, one can be interested in finding groups of objects with a similar pattern in some attributes, and not all of them. Biclustering techniques simultaneously cluster both rows and columns in order to find those groups. The motivation for the work presented in this dissertation comes from the application of biclustering techniques to the metagenomic dataset generated from the worldwide 2014 Ocean Sampling Day event. Since microbial activity is a fundamental component of ocean's biogeochemical cycles, we tried to find geographic niches of certain microbial functions through the application of biclustering techniques. The problem here is how to determine the relevance of a bicluster from a biological and geographical point of view.

We propose a general methodology that evaluates a bicluster considering the relevance of the rows and the relevance of the columns belonging to it. Such relevance is computed relying on a set of indexes defined according to the application domain.

In our case study, the relevance of the rows corresponds to the biological relevance, since the rows of the metagenomic dataset represent the microbial functions. On the other hand, the relevance of the columns corresponds to the geographical relevance, since the columns of the metagenomic dataset represent sampling sites.

We applied our proposed methodology to the case study using ORCA, which is a web application that we developed. Our methodology allowed us to find meaningful biclusters from a biological and geographical point of view. Furthermore, it also allowed us to find interesting relationships, which were unknown so far, between key microbial functions (nitrogen biogeochemistry) within different marine ecosystems. Many of those functional interconnectivities identified with our methodology are relevant from a biological point of view.

Keywords: clustering, biclustering, biclustering validation, metagenomic dataset.

Contents

List of Tables **xv**

List of Figures **xviii**

List of Acronyms **1**

1 Introduction **3**

1.1 Context 3

1.2 Motivation and Problem Statement 5

1.2.1 Ocean Sampling Day 5

1.3 Main Goals 6

1.4 Outline 6

2 Clustering **9**

2.1 Similarity and Dissimilarity 11

2.1.1 Similarity and Dissimilarity Measures 12

2.2 Cluster Definition 15

2.3 Clustering Methods 16

2.3.1 Hierarchical Methods 17
2.3.2 Partitioning Methods 18
2.3.3 Density-based Methods 18
2.3.4 Grid-based Methods 19
2.4 Clustering Structures 20
2.5 Clustering Validation Measures 21
2.5.1 Notation 22
2.5.2 Internal Measures 22
2.5.3 External Validation Measures 25
2.5.3.1 Classification-oriented Measures 25
2.5.3.2 Similarity-oriented Measures 27

3 Biclustering **29**

3.1 Application Example and Notation 31
3.2 Types of Biclusters 33
3.2.1 Constant Biclusters 33
3.2.2 Biclusters with Constant Values on Rows or Columns 35
3.2.3 Biclusters with Coherent Values 36
3.2.4 Biclusters with Coherent Evolutions 37
3.3 Bicluster Structure 37
3.4 Biclustering Methods 39
3.4.1 Constant Biclusters 39

3.4.2 Biclusters with Constant Values on Rows or Columns 39
3.4.3 Biclusters with Coherent Values 41
3.4.4 Biclusters with Coherent Evolutions 47
3.5 Biclustering Validation Measures 48
3.5.1 Non-biological Measures 49
3.5.2 Biological measures 51

4 Domain Oriented Biclustering Validation 53

4.1 Case Study: Presentation 54
4.2 Problem Statement 60
4.3 Methodology 61

5 Case Study 65

5.1 Case study application 66
5.1.1 Collection 66
5.1.2 Rows Indexes 67
5.1.3 Columns Indexes 69
5.1.4 Biclustering Evaluation 71
5.2 ORCA Application 72
5.2.1 Other Features 74
5.3 Results 76

6 Conclusion 79

A Experimental setup **83**

B Biclustering Analysis **85**

B.1 Bicluster 6559 85

B.2 Bicluster 9093 89

Bibliography **95**

List of Tables

4.1 Summarization of environmental variables. 58

5.1 Interesting IPRs. 68

A.1 Experimental parameters. 84

B.1 Summarization of environmental variables. 86

B.2 Summarization of environmental variables. 90

List of Figures

2.1 Different number of clusters. 10

3.1 Examples of different types of biclusters [53]. 34

3.2 Bicluster structure [53]. 40

4.1 Samples distribution. 55

4.2 Number of missing values for each variable. 56

4.3 Distributions of environmental variables. 57

4.4 Significant correlations 59

5.1 Biclustering main menu. 71

5.2 Set weights. 72

5.3 Inspect biclusters. 73

5.4 Biclustering visualization. 74

5.5 Load data. 75

B.1 Expression values of the IPRs along the samples. 86

B.2 Geographical distribution of the sampling sites. 87

B.3 Distribution of the environmental variables. 87

B.4 Significant correlations between IPRs and the environmental variables. 88

B.5 Correlations between IPRs and the environmental variables. 88

B.6 Significant correlations between IPRs. 89

B.7 Expression values of the IPRs along the samples. 90

B.8 Geographical distribution of the sampling sites. 91

B.9 Distribution of the environmental variables. 91

B.10 Significant correlations between IPRs and the environmental variables. 92

B.11 Correlations between IPRs and the environmental variables. 92

B.12 Significant correlations between IPRs. 93

List of Acronyms

DBSCAN Density-Based Spatial Clustering of Applications with Noise 19

DCC Double Conjugated Clustering 45

DNA Deoxyribonucleic Acid 31

DNRA dissimilatory nitrate reduction to ammonia 56, 67, 69, 77, 86

EMBL-EBI European Bioinformatics Institute 55

FLOC Flexible Overlapped biClustering 43, 44

GO Gene Ontology 51, 52, 60

ITWC Interrelated Two-Clustering 45

KEGG Kyoto Encyclopedia of Genes and Genomes 51

MaPle Maximal Pattern-based Clustering 45

Micro B3-IS Micro B3 Information System 54

Micro B3 Marine Microbial Biodiversity, Bioinformatics, Biotechnology 5, 53, 54

OP-Cluster Order Preserving Cluster 47

OPSM order-preserving submatrix 47

OSD Ocean Sampling Day 5, 6, 7, 53, 54, 55, 56, 58, 71, 74, 79

PCA Principal Component Analysis 4, 58

RNA Ribonucleic Acid 31

RSS Residual Sum of Squares 17, 18, 22

STING STatistical INformation Grid-based 19

xMOTIF conserved gene expression motif 48, 66

Chapter 1

Introduction

1.1 Context

Clustering is a core task in data mining. It is used to identify subgroups (clusters) in a given dataset, such that the data objects in a cluster are more similar to each other than the data objects in different clusters. Clustering can be seen as an hypothesis that somehow explains the groupings in the data. Therefore, clustering is widely used in many fields. Some examples of application include social network analysis [58] where clustering is used to find communities in the underlying network. In the field of marketing, clustering is used to organize groups of customers that share similar attributes [68], such as age, gender and income. In the field of biology, clustering is used to detect groups of genes with related expression patterns [10]. The aforementioned list of applications is not exhaustive, and so there are several other fields where clustering is frequently used for data analysis [79], such as astronomy, physics, and medicine.

Although there is an agreement in the literature about the definition of clustering, the notion of what is a cluster is not precisely defined [24], since there are different definitions of what constitutes a cluster and different ways of quantifying the similarity,

or dissimilarity between data objects. Therefore, there are several clustering methods where each one aims to find a certain type of clusters.

Despite the different definitions of what constitutes a cluster, all of the conventional clustering methods have something in common: the data objects in a certain cluster are similar to each other w.r.t. all of the attributes (or features) that describe them. However, this may not be ideal under some contexts. For instance, under the context of gene expression analysis, where the data objects are typically genes and the attributes are conditions, the assumption that all the genes belonging to a cluster behave similarly in all conditions is unrealistic from a biological point of view if we have thousands of conditions, i.e., high dimensional data. In fact, a cellular process may affect a subset of genes only under a subset of conditions. The main challenge for clustering is the fact that different subsets of attributes are relevant for varying clusters. This phenomenon is known as the local feature relevance [45].

Techniques that perform dimensionality reduction, like Principal Component Analysis (PCA), cannot be applied to this case, since they usually compute only one subset of attributes in which the clustering can then be performed. The problem here is that multiple subsets of attributes are required, since different subsets of attributes are relevant for varying clusters, as stated by the problem of local feature relevance. Enumerating all of those subsets and apply clustering to them is computationally unfeasible and is not an option [45].

Biclustering techniques have been applied to gene expression analysis, since they overcome the local feature relevance problem by simultaneously cluster both rows and columns [53]. Biclustering methods aim to discover subsets of data objects with a common pattern under a subset of attributes. In the case of gene expression analysis, biclustering allows us to identify subsets of genes with similar expression patterns under a subset of conditions.

In clustering (one-way clustering or biclustering) the validation of the clusters has a key role, since there is not a definition of what is a good clustering. There are

mainly two kinds of validation measures [34]: internal and external. Internal validation measures evaluate clustering results using only information intrinsic to the data. On the other hand, external validation measures evaluate clustering results using external information about the data.

1.2 Motivation and Problem Statement

Biclustering techniques can be applied to several real-world problems. For instance, they are used in the field of bioinformatics to address the needs for analysing gene expression data. As mentioned in the last section, biclustering allows us to identify subsets of genes with similar expression patterns under a subset of conditions.

Despite several biclustering methods have been presented over the last decade, not a lot of attention has been paid to biclustering evaluation [48]. Moreover, the existing biclustering validation measures are quite incomplete as compared with those of one-way clustering. The reasons behind this are related with some of the biclustering characteristics that we will present later on.

1.2.1 Ocean Sampling Day

Ocean microbial compartment fundamentally influence the ocean's ability to sustain life on Earth [74]. However, despite the clear importance of marine microbes, very little is known about marine microbial diversity. Moreover, the vast majority $(90 - 99\%)$ of marine microorganisms cannot be cultured under standard laboratory conditions. Recent rapid developments in molecular ecology and metagenomics gave rise to a large amount of genetic information that is currently being generated by a number of international projects making significant progress in addressing marine microbial biodiversity in recent years. The Marine Microbial Biodiversity, Bioinformatics, Biotechnology (Micro B3) project [44] investigated global marine microbial biodiversity and their functional capabilities on a single orchestrated Ocean Sampling Day (OSD) event.

The analysis of the metagenomic and metadata datasets generated from the 2014 OSD initiative pose several challenges. Trying to find geographic niches of certain microbial functions is one of them.

Although biclustering techniques allow us to identify several groups (biclusters) with similar expression patterns under a subset of conditions, we still need, in this case study, to assess the relevance of each bicluster from a biological and geographical point of view.

Giving this motivating application, the proposal of this dissertation is to apply biclustering techniques to the metagenomic dataset and validate the resulting biclusters from a biological and geographical point of view.

1.3 Main Goals

This dissertation aims at (i) reviewing the state of art of clustering and biclustering; (ii) applying biclustering techniques to the metagenomic data generated from the 2014 OSD initiative; (iii) present a methodology to evaluate biclusters based on a set of indexes defined according to the application domain; (iv) apply the suggested methodology to the OSD case study.

1.4 Outline

This dissertation is structured in six chapters. The context of each one is described bellow.

Chapter 2 - Clustering presents different definitions of what constitutes a cluster, the different ways for quantifying the similarity between data objects, and the most conventional clustering methods. Besides that we also present the clustering validation problem and some clustering validation measures.

Chapter 3 - Biclustering presents the different definitions of what constitutes a bicluster, the different biclustering structures and the approaches that some of the most prominent biclustering methods use to find the different types of biclusters. This chapter also presents the biclustering validation problem and some of the most well-known biclustering validation measures.

Chapter 4 - Domain Oriented Biclustering Validation Measures presents the OSD initiative as our case study and a brief exploratory data analysis of the metagenomic and metadata datasets. It also presents the problem statement and a methodology that aims to validate biclustering results based on a set of indexes defined taking into account the domain of application.

Chapter 5 - Case study presents the application of the suggested methodology to our case study and the web application ORCA. A biological interpretation for some of the relevant biclusters found is also presented in this chapter.

Chapter 6 - Conclusion presents the final thoughts and outlines possible research directions.

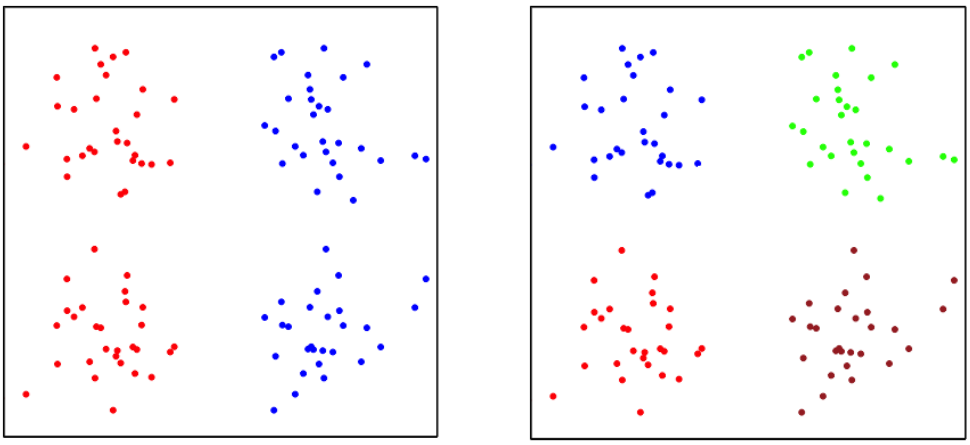
Chapter 2

Clustering

Clustering is the process of partitioning a set of data objects into subsets, also known as *clusters*, where the objects within the same subset are similar to one another and dissimilar to objects in other subsets.

Clustering is widely used for a variety of tasks in many fields. For instance, in biology, clustering can be used to build groups of genes with related expression patterns. In business intelligence, clustering can be used to organize groups of customers that share similar attributes, such as age, gender and income. Clustering has also found many applications in image recognition, where it can be used to group similar images. There are also other clustering applications in Web search, text mining and in many other fields [79].

Although there is an unanimous definition of clustering, the notion of what is a cluster is not precisely defined. As Estivill-Castro [24] describes, *clustering is in part in the eye of the beholder*. To illustrate this concept, Figure 2.1 shows some data objects and two different ways of dividing them into clusters. The color of the objects indicate cluster membership. Figure 2.1(a) and Figure 2.1(b) divide the data into two and four groups, respectively. However, the division of each of the two clusters in Figure 2.1(a) into two subclusters can be and artifact of the human visual system. Figure 2.1 illustrates that the definition of cluster is imprecise and it depends on the nature of data and



(a) Two clusters (b) Four clusters

Figure 2.1: Different number of clusters.

the desired results. Therefore, different clustering algorithms have different definitions of what constitutes a cluster and use different approaches to discover clusters in the most effective way.

Clustering can also be considered as a form of classification, since it divides the objects into groups and each group can be regarded as having a class label. However, clustering does not use previously assigned class labels to group the observations. It derives these labels only from the data. In contrast, classification in the sense of *supervised learning*, consists in the assignment of a class label to a new unlabelled object using a model developed from objects with known class labels. Therefore, in the context of machine learning, clustering is also referred as *unsupervised classification*.

In the section 2.1 we will describe different ways of quantifying the similarity between objects. We also present different types of clusters in the section 2.2, since there are different points of view of what constitutes a cluster. In the section 2.3 we present some of the most prominent clustering methods by categories. These categories differ from each other by the approach that the clustering methods in it use to find clusters in the data. A clustering is the result of applying a clustering method to the data, i.e., a group of clusters. Thus, since there are different clustering methods there are also

different types of clusterings. We describe them in the section 2.4. The purpose of having different types of clusters, clustering methods and types of clusterings is being able to find interesting groups from the data. Therefore, a key question in clustering is how to assess the quality of a clustering. In the section 2.5 we present different clustering validation measures.

Before we start we need to present some notation first. Therefore, we represent a set of n data objects described by m attributes, as an n by m *data matrix* A, where each element a_{ij} is the value of the object a_i in the attribute j. In order to simplify the notation, we will sometimes denote the object a_i by its index i.

Let $prox(i, j)$ be a function that defines a similarity, or dissimilarity, measure between an object i and an object j. The *proximity matrix* P is an n by n matrix where each element p_{ij} is defined as $prox(i, j)$.

2.1 Similarity and Dissimilarity

Data mining problems, such as clustering, outlier detection and classification, require a methodical way (measures) in order to quantify the similarity, or dissimilarity, between data objects.

A similarity measure between two objects, generally returns the value 0 if the objects are unalike. The higher the similarity value, the greater the similarity between two objects. A dissimilarity measure works in the opposite way. It returns the value 0 if the objects are identical. Thus, the higher the dissimilarity value, the more dissimilar the two objects are. Similarity and dissimilarity are then related.

In order to quantify the similarity, or dissimilarity, between objects, some measures calculate the distance between them [56]. However, some other measures, instead of calculating the distances between the objects, use a *similarity function* [20].

The choice of a proper measure can have a great influence in a clustering result [1]. Thus, different measures are required, since there are different data types measured across different data scales.

2.1.1 Similarity and Dissimilarity Measures

Let i, j and k be data objects, such that $i, j, k \in \{1, \ldots, n\}$, and P be an n by n proximity matrix. Similarity and dissimilarity measures must satisfy the following properties, as stated in [37]:

1. (a) For a dissimilarity measure: $p_{ii} = 0$;

   (b) For a similarity measure: $p_{ii} \geq \max p_{ij}$, and thus $p_{ii} = 1$;

2. $p_{ij} = p_{ji}$;

3. $p_{ij} \geq 0$.

In addition, the measures based on distances between objects must satisfy the additional metric properties stated below. Those measures are also know as metric distance measures.

4. $p_{ij} = 0 \Rightarrow i = j$.

5. $p_{ik} \leq p_{ij} + p_{jk}$.

Measures for Numeric Attributes

The Minkowski distance is commonly used to calculate the dissimilarity of objects described by numeric attributes:

$$prox(a_i, a_j) = (|a_{i1} - a_{j1}|^g + |a_{i2} - a_{j2}|^g + \cdots + |a_{im} - a_{jm}|^g)^{\frac{1}{g}}, \qquad (2.1)$$

The Minkowski distance is a generalization of the Euclidean and Manhattan distances. Hence, we achieve the Euclidean distance when $g = 2$ and the Manhattan distance when $g = 1$.

Numeric attributes with different measurement units can heavily affect the clustering results [56]. For instance, if the objects are persons and one attribute is the age and other attribute is the foot length in meters, then the age can have more *weight* or, *importance*, than the foot length. Hence, the data should be *normalized*, i.e., all attributes should have an equal weight.

Measures for Binary Attributes

Calculate the dissimilarity of objects with binary attributes using measures for numeric attributes can be misleading [33]. Hence, there are mainly two measures to assess the dissimilarity between two objects, i and j, with binary attributes: the *simple matching coefficient* and the *jaccard index* [36]. The simple matching coefficient is used when the attributes are symmetric, i.e., both values, 0 and 1, are equally important. Suppose that the objects are student's answers to a True and False test, then, in this case, both matches (00 and 11) are equally important to determine the similarity between student's tests.

Let q be the number of attributes that equal 1 for both objects, t the number of attributes that equal 0 for both objects, r the number of objects that equal 1 for object i but equal 0 for object j, s the number of objects that equal 0 for object i but equal 1 for attribute j and $m = q + t + r + s$ the total number of attributes. Then we define:

$$prox(i, j) = \frac{r + s}{q + t + r + s}, \tag{2.2}$$

as the simple matching coefficient.

However, there are cases where both values are not equally important (*asymmetric*), for example, a disease test where the attributes are conditions like cough and fever.

For those type of attributes, a positive match (11) is more important than a negative match (00). The similarity between objects with asymmetric binary attributes is calculated using the jaccard index:

$$prox(i,j) = \frac{q}{q+r+s}. \tag{2.3}$$

Measures for nominal attributes

There are two common approaches to calculate the dissimilarity between two objects, i and j, when their attributes are nominal. The first one is *simple matching*:

$$prox(i,j) = \frac{m-s}{m}, \tag{2.4}$$

where s is the number of matches (i and j have the same value), and m is the number of attributes describing the objects.

Other approach consists in creating a binary attribute for each value of each nominal attribute. The dissimilarity is calculated as described above. Note that these binary attributes are asymmetric.

Measures for attributes of mixed types

The measures that we have described so far are useful when the attributes of the objects are of the same type. However, there are databases where the objects are described by attributes of mixed types.

The dissimilarity of two objects, i and j, with m attributes of mixed types is defined as:

$$prox(i,j) = \frac{\sum_{k=1}^{m} \delta_{ij}^{(k)} d_{ij}^{(k)}}{\sum_{k=1}^{m} \delta_{ij}^{(k)}}, \tag{2.5}$$

where $\delta_{ij}^{(k)} = 0$ if either the value of the attribute k of the object i, or j, is missing or $a_{ik} = a_{jk} = 0$ and attribute k is asymmetric binary. Otherwise, $\delta_{ij}^{(k)} = 1$. The contribution of attribute k to the dissimilarity between i and j, defined as $d_{ij}^{(k)}$, is calculated according to its type:

- If k is nominal or binary: $d_{ij}^{(k)} = 0$ if $a_{ik} = a_{jk}$ and $d_{ij}^{(k)} = 1$, otherwise.
- If k is numeric: $d_{ij}^{(k)} = \frac{|a_{ik} - a_{jk}|}{\max_h a_{hk} - \min_h a_{hk}}$, where h are all non-missing objects for attribute k.

Cosine Measure

The similarity between objects can be quantified using the angle between them, since they are vectors in a multidimensional space.

For instance, suppose that the objects represent documents where each attribute is the frequency with which a particular word occurs in the document. Our goal is to find groups, i.e., clusters, where the documents in each of them are similar, e.g., all documents belong to the sports category. However, those objects can have thousands of attributes and we should ignore the attributes with the value 0, otherwise most of the documents will be highly similar.

Therefore, the cosine measure is an effective way of quantifying the similarity between those kind of objects [78]:

$$prox(i, j) = \frac{i \cdot j}{\|i\| \cdot \|j\|}. \tag{2.6}$$

2.2 Cluster Definition

As we have already mentioned, the notion of cluster is not precisely defined. However, there are some definitions of what constitutes a cluster [79], which are presented below.

Well-Separated: A cluster is a set of objects where every object in the cluster is more similar to any other object in the cluster than to other object not in the cluster. Sometimes a threshold is used to specify what is the minimum similarity between objects that form a cluster.

Center-based: A cluster is a set of objects where each one is more similar to the center of its cluster than the center of another cluster. The center of a cluster is

typically a centroid or a medoid, i.e., the average of all objects in a cluster or the most representative object in a cluster (the one that has the lowest dissimilarity to all points in the cluster), respectively.

Contiguous: A cluster is a set of objects where each one is more similar to at least one object in the cluster than any other object in a different cluster.

Density-based: A cluster is a connected dense region of objects. Clustering methods that use this definition of cluster are commonly used when the goal is to find irregular or intertwined clusters and when noise and outliers are present.

Conceptual: A cluster is a set of objects that share some property or concept. Suppose that the objects represent persons, then one cluster could be, for example, persons that practise sports. Most conceptual clustering methods generate a hierarchical structure of concepts, and thus, in the example above, other groups under the persons that practise sports could be persons that practise sports with a ball and a group of persons that practise sports without a ball, and so on.

2.3 Clustering Methods

In the section 2.2 we described types of clusters. Those types of clusters are all different and thus there are different clustering methods.

Although it is not possible to categorize all the clustering methods, in this section we provide a brief description of the most prominent categories. The clustering methods can be divided into two main categories [25]: hierarchical and partitioning methods. However, we also include the density-based and grid-based methods as the major fundamental clustering methods [33].

2.3.1 Hierarchical Methods

A hierarchical method creates a hierarchical decomposition of the objects based on the distances between them. In this hierarchy, the clusters are represented using a dendrogram, since there are different clusters, at different levels (distances) of the dendrogram. A clustering of the objects is obtained by cutting the dendrogram at the desired level. These methods do not provide a single partitioning of the data: they provide an hierarchy of clusters that merge with each other at certain distances.

The hierarchy can be obtained following a *top down* or *bottom up* approach. In the bottom up approach, also known as *agglomerative approach*, initially each object is a cluster. Then, the clusters close to one another are merged until all clusters are merged into one.

Conversely, the top down approach, also called *divisive approach*, starts with all objects belonging to the same cluster. Then, it successively splits each cluster into small clusters, until each object forms one cluster.

As we have mentioned earlier, the clusters are merged or divided according to the distances between them. Thus, the hierarchical clustering methods can also be divided according to the manner that the distance is calculated [33]. The single-linkage, complete-linkage and average-linkage are the most prominent ways.

Single-linkage: These methods consider the distance between two clusters to be the minimum distance from any object of one cluster to any object of the other cluster.

Complete-linkage: These methods consider the distance between two clusters to be the maximum distance from any object of one cluster to any object of the other cluster.

Average-linkage: These methods consider the distance between two clusters to be the average distance from any object of one cluster to any object of the other cluster.

There are also other hierarchical clustering methods that do not calculate the distances between clusters as mentioned above. For instance, in the *Ward's method* [39], the criterion chosen to merge the pair of clusters at each step is based on the optimal value of an objective function, such as the Residual Sum of Squares (RSS).

2.3.2 Partitioning Methods

In the partitioning methods each cluster has a *center*, which can be an object or not. The basic idea is, given n objects, construct k partitions, i.e., clusters, where $k \leq n$. These methods start with an initial partitioning and then use and iterative relocation technique that attempts to improve the partitioning by moving objects from one partition to another in a way to minimize a certain error criterion. The error criterion measures the distance of each object to its center. The most prominent one is the RSS:

$$\sum_{i=1}^{k} \sum_{a \in C_i} (a - m_i)^2, \tag{2.7}$$

where each C_i $(1 \leq i \leq k)$ is a cluster, and m_i is the center of the cluster with the index i. This is an optimization problem that is NP-hard [55], since it requires an exhaustive enumeration of all possible partitions in order to find the global minimum of the error criterion chosen. Therefore, the common approach is to search for approximate solutions, i.e., local optimum. This is achieved using heuristics like the greedy approach used by *kmeans* [51], which progressively improves the clustering quality while converging to a local minimum [73].

2.3.3 Density-based Methods

In the density-based methods clusters are defined as areas of high *density* [46].

The general idea behind these methods is to group objects that are within a specified distance from each other and also satisfy a certain density threshold. That distance is also called *radius* and it is useful to define the *neighborhood* of an object. Thus,

the neighborhood of an object a is the space defined by a radius centered at a and its *density* is, typically, the number of objects in it.

The most prominent density-based method is the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [23], where the clusters are defined by all objects that have dense neighborhoods, also called *core objects*, and their neighboors. The radius that defines the neighborhood, ϵ, and the threshold that defines a dense region, *MinPts*, are user-specified parameters.

Partitioning and hierarchical methods find clusters with a spherical shape. Therefore, they are not able to find clusters with an arbitrary shape. They are also sensitive to noise and outliers [33]. The density-based methods find clusters of arbitrary shape and are less sensitive to outliers and noise [23].

2.3.4 Grid-based Methods

Grid-based methods quantize the object space into a finite number of *cells*, that form a grid structure on which all of the clustering calculations are performed.

They take a *space-driven* approach instead of the *data-driven* approach taken by the methods that we have discussed so far: they do not adapt to the distribution of the objects in the space, instead they partition the space into cells regardless the distribution of the objects.

The two most prominent grid-based methods are STatistical INformation Grid-based (STING) [85] and CLIQUE [3]. These methods are summarized below.

In STING the spatial area of the objects is divided into rectangular cells using a hierarchical structure. Each cell at a higher level is divided to form a certain number of cells at a lower level.

Some statistical parameters of each cell are calculated and stored. There are two types of parameters: the attribute-independent (the number of objects in the cell),

and the attribute-dependent for each numerical attribute (the mean, the minimum value, the maximum value, the standard deviation and the distribution). After the computation of the parameters, STING generates a hierarchical structure of the grid cells in order to represent those statistical parameters at different levels. Then, based on this structure, it uses a top down approach to answer queries, i.e., find groups of objects that satisfy some condition.

CLIQUE searches for clusters in dense subspaces of the data. It starts by partitioning each dimension into a specified number of intervals that have the same length, also called cells. Then it identifies the dense cells in all subspaces. A cell is dense if the number of objects within it exceeds a certain density threshold. After these steps, CLIQUE uses the dense cells in each subspace to assemble clusters.

2.4 Clustering Structures

Different clustering methods are able to return different types of clusterings. Therefore, the clustering can be *exclusive*, *overlapping* or *fuzzy*.

The clustering is *exclusive* when an object is assigned to a single cluster. However, there are clustering methods that assign an object to multiple clusters. These clusterings are known as *overlapping* or *non-exclusive*.

In a *fuzzy* clustering, every object belongs to every cluster with a certain membership degree, or probability, that is between 0 (does not belong) and 1 (absolutely belongs). The total sum of these probabilities, for each object, must equal 1.

A clustering can also be *complete* or *partial*. It is *complete* when it assigns every object to a cluster. However, there may be objects that do not belong to any cluster, such as outliers, and in that case it can be a better option not to assign them to any cluster. Thus, a clustering is *partial* when there is at least one object that does not belong to any cluster.

2.5 Clustering Validation Measures

There is no definition of what is a good clustering [12]. As we have seen in the previous sections there are many types of clusters and clustering methods which have their own definition of cluster. Those methods employ different approaches and use different similarity measures. Hence, different clustering results can be returned for the same data, even if that data has no cluster structure, i.e., the data is randomly distributed [79]. This is why cluster validation has a key role in cluster analysis.

The following problems are within the domain of cluster validation:

1. Figure out if a non-random structure actually exists in the data, i.e., the clustering tendency.
2. Figure out the correct number of clusters.
3. Evaluate the goodness of a clustering structure without relying on external information.
4. Evaluate the goodness of a clustering structure relying on external information, such as class labels.
5. Compare two sets of clusters in order to find which is the best one.

There are mainly two types of validation measures: internal and external measures [34]. They are also commonly referred as indexes. External measures are distinguished from internal measures by the use of prior information about known class labels.

In this section we will describe some internal and external validation measures and the key concepts behind them. However, those measures are useful just for partitioning clustering, where only the final result of partitioning is evaluated. For hierarchical clustering the result of each step of the agglomerative or divisive process is evaluated and therefore, different measures as the *cophenetic correlation coefficient* [76] are required. Furthermore, we will only consider exclusive clustering.

2.5.1 Notation

Let $D = \{C_1, C_2, ...C_k\}$ be a set of k clusters, i.e., a clustering result, where $C_i \cap C_z = \varnothing$ for all $i \neq z$ and $\sum_{i=1}^{k} |C_i| = n$. Also, let c_i be the center of cluster C_i and c the center of all objects.

For the external validation measures, where class labels are available, let C_{ij} be the set of objects in cluster C_i which belong to class j, L the total number of classes and m_j the number of objects in class j ($1 \leq j \leq L$).

The function $prox(a, b)$ represents a dissimilarity measure based on the distance between the object a and the object b.

2.5.2 Internal Measures

Internal validation measures evaluate clustering results using only information intrinsic to the data.

As the goal of clustering is to find groups, i.e., clusters, where the dissimilarity between elements within the same cluster is minimized while the dissimilarity of elements within different clusters is maximized, most of the internal validation measures are based on the following two concepts [79]:

Cohesion (compactness): This is a quantification of how similar are the objects within the same cluster. The cohesion can be measured using the RSS defined as:

$$RSS(D) = \sum_{x \in C_i} prox(x, c_i), \tag{2.8}$$

where $prox(x, c_i)$ represents the Euclidean distance and a low value indicates a better cohesion. Besides the RSS, the cohesion can also be measured by the maximum or the average pairwise distance, and the maximum or the average center-based distance, between all objects in the cluster.

Separation: This is a quantification of how dissimilar or "well-separated" a cluster is from the others. The separation can be measured in multiple ways, such as the pairwise distance between clusters centers and the pairwise minimum distance distance between objects in different clusters.

Therefore, internal validation measures combine different versions of these concepts to assess the quality of a clustering. There are also other measures based on different approaches, such as the proximity matrix [79]. However, we will describe only the most prominent measures based on cohesion and separation.

Dunn's index The *Dunn's index* [21] can be formulated as:

$$DI(D) = \frac{\min_{a \in C_i, b \in C_j, 1 \leq i \neq j \leq k} \ prox(a,b)}{\max_{a,b \in C_i, 1 \leq i \leq k} \ prox(a,b)}. \tag{2.9}$$

It is simply the ratio between the minimum pairwise distance of objects belonging to different clusters and the maximum pairwise distance of objects belonging to the same cluster.

Since the numerator increases as clusters are separated from each other and the denominator decreases as the cohesion increases, we seek the maximization of the Dunn's index.

Davies-Bouldin index The *Davies-Bouldin* [19] index is defined as:

$$DB(D) = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{prox(c_i, c_j)} \right), \tag{2.10}$$

where σ_i is the average distance of all elements in cluster C_i to its centroid c_i:

$$\frac{1}{|C_i|} \sum_{a \in C_i} prox(a, c_i).$$

The numerator decreases as the cohesion increases, and the denominator increases as clusters are separated from each other. Thus, we seek the minimization of the Davies-Bouldin index.

Calinski and Harabasz The *Calinski and Harabasz* [16] index is given by:

$$CH(D) = \frac{\frac{1}{k-1}\sum_{i=1}^{k}|C_i|\,prox(c_i, c)}{\frac{1}{n-k}\sum_{i=1}^{k}\sum_{a\in C_i}prox(a, c_i)}. \tag{2.11}$$

In this case, the numerator increases as clusters are separated from each other and the denominator decreases as cohesion increases. Therefore, a higher value of the Calinski and Harabasz index indicates a better clustering quality than a lower one.

Silhouette Coefficient The *silhouette coefficient* [70] is widely used an one of the most popular measures. It quantifies how well each object lies within its cluster using the ideas of cohesion and separation. The silhouette coefficient is defined as:

$$s(a) = \frac{\mu(a) - \alpha(a)}{\max(\mu(a), \alpha(a))}, \tag{2.12}$$

where $\mu(a)$ is the minimum average distance from object a to all clusters to which a does not belong (the average distance from a to a cluster is the average distance between a and all the objects in that cluster). The cluster with the minimum average dissimilarity is also known as the *neighbouring cluster* of a because it is the cluster to which a is closer (besides its own cluster). Let $a \in C_i$ $(1 \leq i \leq k)$, then

$$\mu(a) = \min_{C_j:1\leq j\leq k, j\neq i}\left(\frac{\sum_{b\in C_j}prox(a, b)}{|C_j|}\right),$$

and $\alpha(a)$ is the average distance between object a and all other objects in the cluster to which a belongs:

$$\alpha(a) = \frac{1}{|C_i| - 1}\sum_{b\in C_i, b\neq a}prox(a, b).$$

As $\mu(a)$ is a measure of how dissimilar a is to its neighbouring cluster, the largest the value of $\mu(a)$ the better the separation is. Furthermore, $\alpha(a)$ is a measure of how dissimilar a is to its own cluster, and thus a small value indicates a better cohesion than a higher one.

The values of the silhouette coefficient are between -1 and 1, where 1 is the preferable case. When $s(a) > 0$, the object a is closer to its cluster than its neighbouring cluster. Contrarily, when $s(a) < 0$, the object a is closer to its neighbouring cluster than its own cluster.

To measure the quality of a cluster we can compute the average silhouette coefficient of all objects in that cluster. Likewise, to measure the quality of a clustering we can compute the average silhouette coefficient of all objects.

2.5.3 External Validation Measures

External validation measures evaluate clustering results using external information about the data - typically class labels for the objects. The idea is to compare a clustering result against a known set of class labels, also called *ground truth*, in order to assess the consensus between the two.

We consider two different groups of external validation measures [79]: *classification-oriented* and *similarity-oriented* measures. The measures in the first group are based on the measures used for assessing the quality of a classification model, such as the Entropy, the Purity and the F-measure. The goal of these measures is to evaluate the prevalence of a single class in a certain cluster. In its turn, the similarity-oriented measures are based on the similarity measures for binary attributes, such as the jaccard index.

2.5.3.1 Classification-oriented Measures

In order to measure the quality of a classification model, we measure the degree to which predicted class labels match the true class labels.

In clustering, instead of having predicted class labels we have cluster labels. Hence, the measures used to evaluate the performance of a classification model can be used to evaluate a clustering result where the class labels are known.

Entropy The *entropy* is the degree to which each cluster consists of objects of a single class. Let pr_{ij} be the probability that a member of a cluster with index i belongs to class j defined as:

$$pr_{ij} = \frac{|C_{ij}|}{|C_i|}.$$

Thus, the entropy of a cluster with index i $(1 \leq i \leq k)$ is defined as:

$$e(C_i) = -\sum_{j=1}^{L} pr_{ij} \log_2 pr_{ij}. \tag{2.13}$$

The total entropy of a clustering result is the sum of the entropies of each cluster weighted by the size of each cluster, and it should be minimized:

$$e(D) = \sum_{i=1}^{k} e_i \frac{|C_i|}{n}. \tag{2.14}$$

Purity The *purity* is another measure of the degree to which each cluster consists of objects of a single class.

The purity of a cluster with index i is defined as:

$$pur(C_i) = \max_j \ pr_{ij}. \tag{2.15}$$

Likewise, the overall purity of a clustering result is defined as:

$$purity(D) = \sum_{i=1}^{k} pur_i \frac{|C_i|}{n}. \tag{2.16}$$

Precision The *precision* of a cluster with index i w.r.t a certain class j is the fraction of objects within C_i that belong to class j.

It is the same as pr_{ij} and it is defined as *precision(i,j)*.

Recall Recall is the fraction of objects with class j in a cluster with index i:

$$recall(i, j) = \frac{|C_{ij}|}{m_j}. \tag{2.17}$$

F-measure *F-measure* is a weighted average of precision and recall, and measures the extent to which a cluster contains only objects of a particular class and all objects of that class. The F-measure of a cluster with index i w.r.t class j is defined as:

$$F_1(i,j) = \frac{2 \cdot precision(i,j) \cdot recall(i,j)}{precision(i,j) + recall(i,j)}. \tag{2.18}$$

2.5.3.2 Similarity-oriented Measures

These measures are used to evaluate the agreement between a clustering result and the true class labels.

Calculating the similarity between binary attributes is not the same as calculating the quality of a clustering result. Thus, a change in notation is required:

f_{00} = number of pairs of objects having a different class and a different cluster,

f_{01} = number of pairs of objects having a different class and the same cluster,

f_{10} = number of pairs of objects having the same class and a different cluster,

f_{11} = number of pairs of objects having the same class and the same cluster.

The most prominent similarity oriented measures are the *jaccard index* and the *Rand statistic* [69]:

$$RandStatistic(D) = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}}, \tag{2.19}$$

$$jaccardIndex(D) = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}. \tag{2.20}$$

The jaccard index does not use the value of f_{00}, i.e., only the number of pairs of objects having the same class and the same cluster are taken into account.

Chapter 3

Biclustering

So far, we have presented clustering as problem where the main goal is to group similar objects w.r.t. all attributes.

However, there are other formulations of clustering problems. For instance, one can be interested in finding groups of objects with a common pattern under a subset of attributes. In this case, instead of grouping the objects based on the similarity between them, the idea is to find *patterns* on the data, i.e., submatrices of the data matrix. Thus, a cluster is now defined as a subset of objects and a subset of attributes.

These clusters are known as *biclusters* and, in order to find them we need new methods to group both objects and attributes simultaneously. These types of clustering methods belong to the category of *biclustering*.

Besides the property that a bicluster is a subset of objects and attributes, there are also other properties, such as an object may belong to multiple biclusters, or does not belong to any, and an attribute can also belong to multiple biclusters, or does not belong to any.

In the section 3.1 we introduce biclustering by means of one application example: analysis of gene expression data, since biclustering is widely used in the field of bioinformatics [61, 80]. The term was first introduced by Mirkin [57], although the

technique was introduced much earlier by Hartigan [35]. Cheng and Church [18] applied the first biclustering algorithm in the analysis of gene expression data.

There are different types and structures of biclusters. This is why several methods, that rely on different heuristic approaches, have been developed. Almost all biclustering methods use heuristic approaches, since it has been proven that the complexity of the task of finding all the significant biclusters, although it may depend on the exact problem formulation and, more specifically, on the merit function used to evaluate the quality of a given bicluster, is NP-complete [80]. Finding a maximum size bicluster in a binary dataset is equivalent to finding the maximum edge biclique in a bipartite graph which is known to be a NP-complete problem [64]. Hence, biclustering methods limit the search space using heuristic approaches.

In the sections 3.2 and 3.3 we present different types and structures of biclusters, respectively, based on the work of Madeira and Oliveira [53].

Biclustering methods can be classified based on the types, structure of the identified biclusters and the heuristic approach. However, in the section 3.4, we just provide a brief explanation of some biclustering methods according to the different biclustering types as Kriegal et al. [45], since a detailed description of the several existing methods is beyond the scope of this dissertation. A more exhaustive covering of biclustering algorithms can be found in other surveys [22, 53, 81].

Despite the fact that biclustering methods are able to find useful groups that, in some cases, may not be obtained with traditional clustering methods, biclustering still belongs to the category of unsupervised learning problems. Hence, assessing the quality, or the relevance of a biclustering result is still one of the most important open issues [53]. Besides that, there are far less studies and measures to validate biclustering results, in comparison with the traditional clustering approaches that have been studied actively over the years. The main focus of this thesis is the development of a new approach to validate biclustering results.

In the section 3.5 we describe some biclustering validation measures. Similarly as in clustering, biclustering validation measures can be divided into two groups: internal and external validation measures.

3.1 Application Example and Notation

Gene Expression Data *Genes* are segments of the Deoxyribonucleic Acid (DNA) that specify all proteins and functional Ribonucleic Acid (RNA) chains. They hold the information to build and maintain the cells of a living organism. The synthesis of a functional gene product (RNA or protein) is known as *gene expression*.

DNA chips, also known as *DNA microarrays*, and other techniques enable us to measure the expression level of a large number of genes, perhaps all genes of an organism, in a number of different experimental conditions [7]. Such conditions may be different time points in an experiment, samples from different organs or samples from different tissues, e.g., cancerous or healthy tissues. Thus, the gene expression data is commonly represented as a data matrix where each gene corresponds to one row (object) and each sample, or condition, to one column (attribute). Each element of this matrix represents the expression level of a gene under a specific condition.

Clustering techniques have been applied to gene expression data in two dimensions: the *gene dimension* and the *condition dimension*. When analysing in the gene dimension, each gene is an object and the conditions are treated as attributes. The main goal is to find groups of genes that express themselves similarly in all the conditions. Conversely, when analysing in the condition dimension, each condition is an object and the genes are treated as attributes. This is useful to find patterns of conditions or cluster them into groups.

However, applying traditional clustering methods to gene expression data has several drawbacks. The first one, and maybe the most important, is the assumption that all the genes belonging to a cluster are similar with each other in all conditions. In fact,

a cellular process may affect a subset of genes only under certain conditions. Thus, we need methods to identify similar groups of genes under a subset of conditions. This can be solved by biclustering techniques that perform simultaneous clustering on both rows and columns, instead of clustering these two dimensions separately. While traditional clustering methods derive a *global model*, i.e., the similarity in a gene cluster is measured w.r.t. all conditions, biclustering methods are able to derive a *local model* where genes in the same bicluster share an expression pattern under a subset of conditions.

Besides deriving a local model from the data, biclustering is also useful to tackle problems related with *high dimensional data* [45].

Notation We consider a data matrix A with n rows and m columns as being defined by its sets of rows, $X = \{x_1, \ldots, x_n\}$, and its sets of columns, $Y = \{y_1, \ldots, y_m\}$. The data matrix A is denoted as (X, Y). Let $I \subseteq X$ and $J \subseteq Y$, we define $A_{IJ} = (I, J)$ as the submatrix that contains only the objects a_{ij} corresponding to the rows I and columns J, such that $i \in I$ and $j \in J$. A bicluster is a submatrix $A_{IJ} = (I, J)$, where $I = \{i_1, \ldots, i_k\} \subseteq X$ and $J = \{j_1, \ldots, j_s\} \subseteq Y$.

Through this notation it is possible to define the mean values of rows, of columns and of a certain submatrix (bicluster). Thus, the mean of the ith row in the bicluster (I, J) is given by:

$$a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{ij}. \tag{3.1}$$

The mean of the jth column in the bicluster (I, J) is given by:

$$a_{Ij} = \frac{1}{|I|} \sum_{i \in I} a_{ij}. \tag{3.2}$$

The mean of all elements in the bicluster (I, J) is given by:

$$a_{IJ} = \frac{1}{|I|\,|J|} \sum_{i \in I,\, j \in J} a_{ij} \tag{3.3}$$

$$= \frac{1}{|I|} \sum_{i \in I} a_{iJ} \tag{3.4}$$

$$= \frac{1}{|J|} \sum_{j \in J} a_{Ij}. \tag{3.5}$$

3.2 Types of Biclusters

According to Madeira and Oliveira [53] there are four different categories of biclusters: constant biclusters, biclusters with constant values on either columns or rows, biclusters with coherent values and biclusters with coherent evolutions.

In the biclusters belonging to the first three categories, it is possible to observe patterns w.r.t. the values in the data matrix. These patterns can be observed in the rows, in the columns or in both dimensions of the data matrix. In the biclusters belonging to the fourth category it is possible to observe a pattern based on the behaviour, regardless of the exact numeric values. Thus, biclusters with coherent evolutions view the elements in the data matrix as symbols.

3.2.1 Constant Biclusters

A *constant bicluster*, Figure 3.1(a), is a submatrix (I, J) where all entries have the same value. Formally, a *perfect constant bicluster* consists of objects sharing an identical value, μ, on a subset of attributes, for all $i \in I$ and $j \in J$:

$$a_{ij} = \mu. \tag{3.6}$$

However, in real data, constant biclusters are usually masked by noise. With this, the values in a constant bicluster, a_{ij}, are generally presented as $\eta_{ij} + \mu$, where η_{ij} is the noise associated with the value μ of a_{ij}. The merit function used to compute and evaluate constant biclusters is generally the variance.

A perfect constant bicluster is a submatrix where the variance is equal to zero. Therefore, searching for perfect constant biclusters on the data matrix $A = (X, Y)$

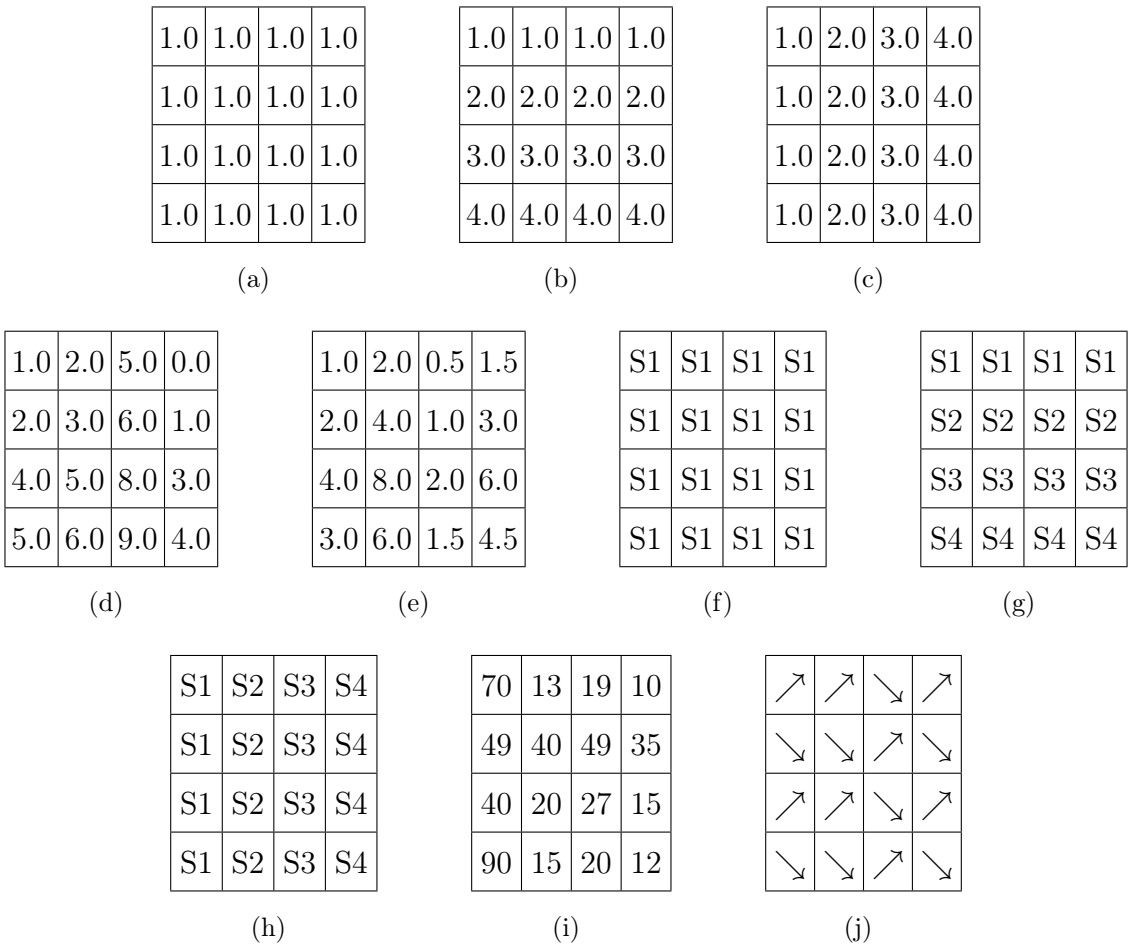


Figure 3.1: Examples of different types of biclusters [53].
(a) Constant bicluster, (b) constant rows, (c) constant columns, (d) coherent values (addictive model), (e) coherent values (multiplicative model), (f) overall coherent evolution, (g) coherent evolution on the rows, (h) coherent evolution on the columns, (i) coherent evolution on the columns, and (j) coherent sign changes on rows and columns.

will probably return $|X| \cdot |Y|$ biclusters, i.e., each element a_{ij} of the data matrix is a perfect bicluster. As such, the methods used to find constant biclusters implement some techniques, that we will describe later on, in order to avoid this partitioning of the data.

3.2.2 Biclusters with Constant Values on Rows or Columns

This type of biclusters raises great practical interest. Using the application example of the gene expression analysis, a *bicluster with constant rows*, Figure 3.1(b), identifies a subset of genes with similar expression values under a subset of conditions. In this case, the expression levels differ from gene to gene, but they only differ from each other by some adjustment value, α_i, associated with each row $i \in I$. A *perfect bicluster with constant rows* is a submatrix (I, J) where each value, a_{ij}, can be obtained by:

$$a_{ij} = \mu + \alpha_i, \tag{3.7}$$

$$a_{ij} = \mu \times \alpha_i, \tag{3.8}$$

where μ is the common value in the bicluster. Each value can be obtained in an additive (equation 3.7) or multiplicative way (equation 3.8).

A *bicluster with constant columns*, Figure 3.1(c), in the context of the gene expression analysis, is a subset of conditions within which a subset of genes have similar expression values. In these type of biclusters, the expression levels differ from condition to condition, but they only differ from each other by some adjustment value, β_j, associated with each column $j \in J$. A *perfect bicluster with constant columns* is a submatrix (I, J) where each value, a_{ij}, can be obtained by:

$$a_{ij} = \mu + \beta_j, \tag{3.9}$$

$$a_{ij} = \mu \times \beta_j. \tag{3.10}$$

Similarly, each value can be obtained in an additive (equation 3.9) or multiplicative way (equation 3.10).

Using the variance as the merit function in order to find these kind of biclusters is not enough. Hence, the simplest approach consists in first normalizing the rows (columns) of the data matrix using the row mean (column mean). The row (column) normalization transforms biclusters with constant values on the rows (columns) into constant biclusters. There are also other approaches to find these kind of biclusters, since perfect biclusters with constant rows (columns) are hard to find in real data due to the noise. We will describe those approaches later on.

3.2.3 Biclusters with Coherent Values

One can be interested in finding more complex patterns on the data. Hence, more sophisticated approaches have been developed to find *biclusters with coherent values* on both rows and columns. These types of biclusters can be described by a combination of the equations 3.7 and 3.9, for the additive model, and the equations 3.8 and 3.10, for the multiplicative model.

A *perfect bicluster with coherent values* is a submatrix (I, J) where each value, a_{ij}, can be obtained by:

$$a_{ij} = \mu + \alpha_i + \beta_j, \tag{3.11}$$

$$a_{ij} = \mu' \times \alpha'_i \times \beta'_j. \tag{3.12}$$

The equation 3.11 describes an *additive model*, where α_i is an adjustment value for row $i \in I$, and β_j is an adjustment value for column $j \in J$. In this additive model both adjustment values are simultaneously used to adjust the mean value μ to a certain value in row i and column j. Furthermore, the equations 3.7 and 3.9 are special cases of the equation 3.11 when $\beta_j = 0$ or $\alpha_i = 0$, respectively. Figure 3.1(d) depicts a bicluster with coherent values (additive model).

Other approaches assume a *multiplicative model* described by the equation 3.12, that is equivalent to the additive model in the equation 3.11 when $\mu = log(\mu')$, $\alpha_i = log(\alpha'_i)$ and $\beta_j = log(\beta'_j)$. In this model, each element a_{ij} is the result of the product between

the common value, μ', an adjustment value for row $i \in I$, α'_i, and an adjustment value for column $j \in J$, β'_j. Figure 3.1(e) depicts a bicluster with coherent values (multiplicative model).

3.2.4 Biclusters with Coherent Evolutions

Besides biclusters with coherent values there are also *biclusters with coherent evolutions*, as the ones depicted in Figure 3.1(f), 3.1(g), 3.1(h), 3.1(i) and 3.1(j). In this type of biclusters it is possible to observe coherent evolutions along the rows and/or columns of the data matrix regardless of their exact values. These coherent evolutions are described by a subset of rows and/or columns where the values within it change in the same direction, or by a subset of rows and/or columns where the values within it are in the same state. Those states can be obtained through the discretization of the values in the data matrix into levels.

Finding biclusters with coherent evolutions, in the context of the gene expression data example, can be useful when the goal is to find a subset of genes that are upregulated or downregulated under a subset of conditions, regardless of the expression values.

3.3 Bicluster Structure

As mentioned above, besides the variations in the types of the biclusters there are also other kinds of variations, such as the size and the position of the biclusters in the data matrix. Although there are methods that aim to find a *single bicluster*, Figure 3.2(a), several approaches assume the existence of several biclusters. For those approaches, different bicluster structures can be obtained [53]:

1. Exclusive row and column biclusters;

2. Nonoverlapping biclusters with checkboard structure.

3. Exclusive-rows biclusters.

4. Exclusive-columns biclusters.

5. Nonoverlapping biclusters with tree structure.

6. Nonoverlapping nonexclusive biclusters.

7. Overlapping biclusters with hierarchical structure.

8. Arbitrarily positioned overlapping biclusters.

It is possible to represent the values of a matrix through an image. In order to do that we set a color for each value a_{ij}. Then, if we try to reorder the data matrix with the goal of finding groups with similar rows and similar columns, we will have as a result some K rectangular blocks (each one corresponding to one bicluster) on the diagonal of our visual representation of the data matrix. As expected, each block will be uniformly colored. Figure 3.2(b) describes this reordering, where the blocks are *exclusive row and column biclusters* (every row and every column belongs exclusively to one of the K biclusters).

However, the reordering that we have described is not common in real data. Most of the times, rows and columns may belong to more than one bicluster. Thus, in a *checkerboard structure*, depicted in Figure 3.2(c), we allow the existence of K *nonoverlapping and nonexclusive biclusters.*

There are also other structures, where rows can only belong to one bicluster while columns can belong to several biclusters or, conversely, rows can belong to several biclusters while columns can only belong to one bicluster. These kind of structures are depicted in Figures 3.2(d) and 3.2(e), and are known as *exclusive rows biclusters* and *exclusive columns biclusters*, respectively.

The structures that we have presented so far assume that the biclusters are *exhaustive*, i.e., every row and every column belong to at least one bicluster. Figures 3.2(f) and 3.2(g) also depict *exhaustive bicluster structures.* However, it may be useful to

consider that some rows and columns may not belong to any bicluster. Besides that, all of those structures consider that there is no overlap between biclusters, which, once again, may be idealistic. In Figure 3.2(h) we can observe *overlapping biclusters with a hierarchical structure* that requires that either the biclusters are disjoint or that one includes the other. A more general bicluster structure, depicted in Figure 3.2(i), allows the existence of overlapping, nonexclusive and nonexhaustive biclusters positioned arbitrarily.

3.4 Biclustering Methods

3.4.1 Constant Biclusters

As we have mentioned in the section 3.2.1, the merit function used to compute and evaluate constant biclusters is, in general, the variance. Thus, a perfect bicluster is a submatrix with variance equal to zero.

Block Clustering, introduced by Hartigan [35], is a partition based method that splits the original data matrix into a set of submatrices and uses the variance to evaluate the quality of each bicluster (I, J):

$$VAR(I, J) = \sum_{i \in I,\, j \in J} (a_{ij} - a_{IJ})^2. \tag{3.13}$$

The data matrix is split recursively into two partitions. At each step, the split that maximizes the reduction in the overall variance of all biclusters is chosen. The splitting stops when the reduction in variance due to further splitting is less than that expected by chance.

3.4.2 Biclusters with Constant Values on Rows or Columns

The straightforward approach to identify these type of biclusters consists in a normalization step that enable us to find constant biclusters. Getz et al. [27] presented

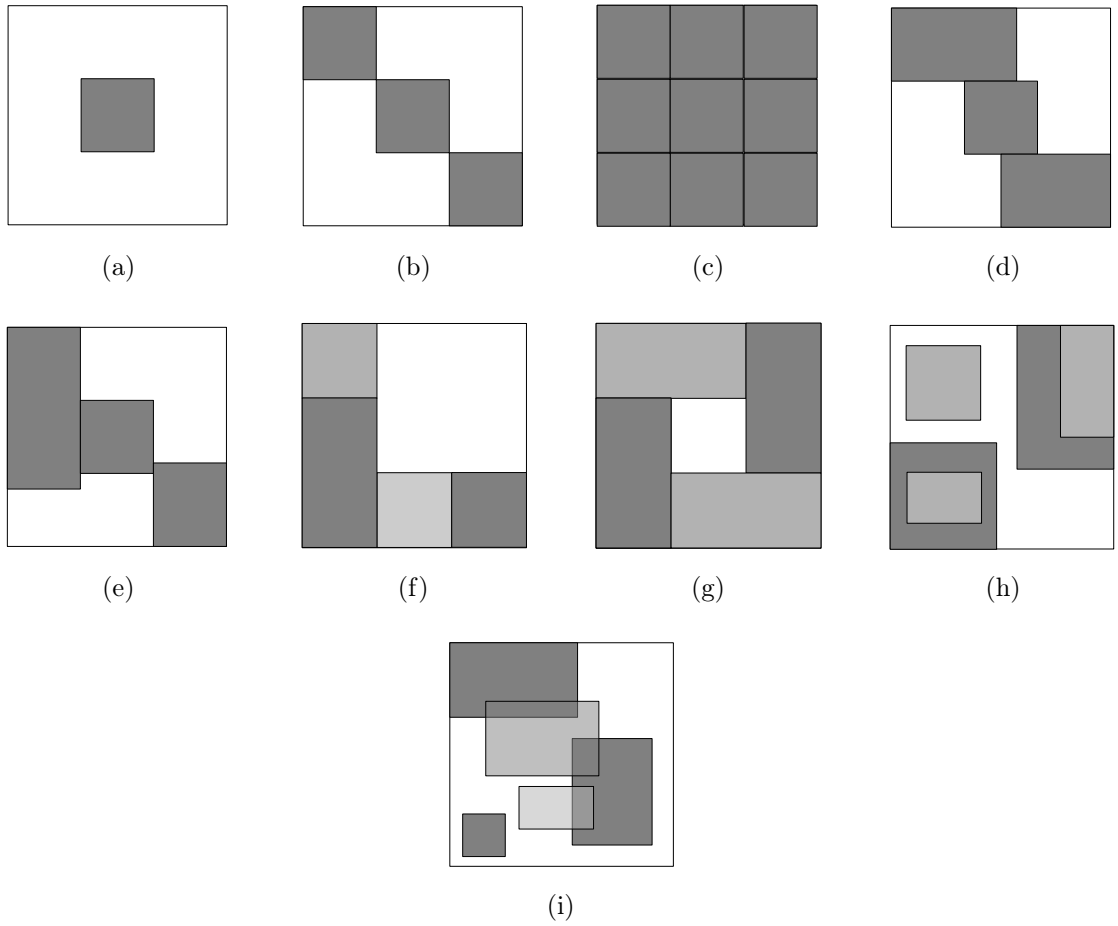


Figure 3.2: Bicluster structure [53].
(a) Single bicluster, (b) exclusive row and column biclusters, (c) checkerboard structure, (d) exclusive rows biclusters, (e) exclusive columns biclusters, (f) nonoverlapping biclusters with tree structure, (g) nonoverlapping nonexclusive biclusters, (h) overlapping biclusters with hierarchical structure, and (i) arbitrarily positioned overlapping biclusters.

a method based on a normalization step that is able to find biclusters with constant rows or constant columns.

There are also other methods that do not rely on the normalization step. For instance, Califano et al. [15] presented a method where the goal is to find δ-valid ks-patterns. A δ-valid ks-pattern is defined as being a subset of rows, I, with size k, and a subset of columns, J, with size s, where the difference between the maximum and the minimum value of each row, along the subset of columns J, is less than δ. Thus, for each row $i \in I$:

$$max(a_{ij}) - min(a_{ij}) < \delta, \forall j \in J. \tag{3.14}$$

A δ-valid ks-pattern is maximal if it cannot be extended into a δ-valid $k's$-pattern, $k' > k$, by adding rows to its row set and if it cannot be extended to a δ-valid ks'-pattern, $s' > s$, by adding columns to its column set. The goal is to find maximal δ-valid ks-patterns. This approach does not aim to find *perfect* biclusters with constant rows or constant columns, since it considers the possible existence of noise.

3.4.3 Biclusters with Coherent Values

Cheng and Church [18] introduced a method that relies on a similarity score, called *mean squared residue*, denoted by H, in order to find coherent biclusters. A submatrix (I, J) is a *δ-bicluster* if its mean squared residue is below a certain threshold, δ, i.e., $H(I, J) < \delta$ where $\delta \geq 0$. The goal of this method is to find large and maximal δ-biclusters. If we set $\delta = 0$ we will be seeking for biclusters where each row/column, or both rows and columns, have an absolutely consistent bias. Those biclusters are known as *perfect δ-biclusters*. The biclusters depicted in Figure 3.1(b) and Figure 3.1(c), where the values of each row or column, respectively, can be obtained by shifting the values of other rows or columns by a common value, are examples of perfect biclusters. When dealing with perfect biclusters, the relative bias of row i w.r.t the other rows is given by $a_{iJ} - a_{IJ}$. Likewise, the relative bias of column j w.r.t the other columns is given by $a_{Ij} - a_{IJ}$. The value a_{ij} is then given by the sum of a

row-constant, a column-constant and an overall constant value:

$$a_{ij} = a_{iJ} - a_{Ij} - a_{IJ}. \tag{3.15}$$

From this, the equation 3.11 can be obtained by considering $\mu = a_{IJ}$, $\alpha_i = a_{iJ} - a_{IJ}$ and $\beta_j = a_{Ij} - a_{IJ}$.

However, the value of a_{ij} cannot be obtained from the equation 3.15 if the data contains noise. In this case, the δ-biclusters are not perfect and we need to measure the difference between the actual value of an element and its expected value (predicted using the equation 3.15). This difference is known as *residue*. Therefore, the value of an element a_{ij} in a nonperfect bicluster is given by:

$$a_{ij} = r(a_{ij}) + a_{iJ} + a_{Ij} - a_{IJ}, \tag{3.16}$$

where $r(a_{ij})$ is the residue defined as:

$$r(a_{ij}) = a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}. \tag{3.17}$$

According to Cheng and Church, the *mean squared residue* of the bicluster (I, J) is given by:

$$H(I, J) = \frac{1}{|I|\,|J|} \sum_{i \in I,\, j \in J} r(a_{ij})^2. \tag{3.18}$$

As we have mentioned above, the goal of the method proposed by Cheng and Church is to find maximal δ-biclusters. However, this task can be computationally expensive and thus, a heuristic greedy approach that returns a local optimum is used. The method can be summarized in two phases [33]: deletion phase and addition phase.

Deletion phase: In this phase we iteratively remove rows and columns from the whole matrix as long as the mean squared residue is greater than a certain δ. At each iteration we remove the row or the column with the largest mean squared residue. The mean squared residue for each row i is given by:

$$d(i) = \frac{1}{|J|} \sum_{j \in J} r(a_{ij})^2. \tag{3.19}$$

Similarly, the mean squared residue for each column j is given by:

$$d(j) = \frac{1}{|I|} \sum_{i \in I} r(a_{ij})^2. \tag{3.20}$$

The final result of this phase will be a δ-bicluster that may not be maximal. A δ-bicluster is not maximal when there is the possibility of adding more rows and/or columns to it and still maintain the δ-bicluster property, i.e., $H(I, J) < \delta$.

Addition phase: In this phase we interactively expand the δ-bicluster returned by the deletion phase, as a consequence of it may not be maximal. Once again, the δ-bicluster property should be respect. At each iteration we add a row or a column, not involved in the bicluster, which has the lowest mean squared residue.

This method is not able to find more than one δ-bicluster at a time. Therefore, it is repeated K times in order to find K δ-biclusters. Once a bicluster is found, the values of the elements in it, are masked by random numbers. With this, elements belonging to a certain bicluster will seldom contribute to any future bicluster. This means that it is unlikely that the resulting set of K δ-biclusters have overlaps. Besides that, this method assumes that there are no missing values in the data, and thus, missing values are replaced by random numbers. It is expected that these random values do not form a recognizable pattern.

The original δ-bicluster model proposed by Cheng and Church is not able to find certain kinds of biclusters since:

1. It finds one bicluster at a time, then the values of the elements in that bicluster need to be masked in order to find further biclusters.

2. Less accurate results can be returned due to the masking.

3. The masking prevents simultaneous overlapping of rows and columns.

4. It cannot handle with missing values.

The Flexible Overlapped biClustering (FLOC) method proposed by Yang et al. [95,96] takes into account the aspects mentioned above in order to provide an alternative to the original δ-bicluster model. To be able to deal with missing values, FLOC introduces an *occupancy threshold* υ and defines a δ-bicluster with υ occupancy as a submatrix (I, J), where for each row $i \in I$, $\frac{|J'_i|}{|J|} > \upsilon$, and for each column $j \in J$, $\frac{|I'_j|}{|I|} > \upsilon$. In this case $|J'_i|$ and $|I'_j|$ are the number of non-missing values on row i and column j, respectively. The volume of a bicluster, i.e., the total number of non-missing values, is defined as υ_{IJ}. Through this notation we can redefine a_{iJ}, a_{Ij}, a_{IJ}, $r(a_{ij})$ and $H(I, J)$ in the equations 3.1, 3.2, 3.3, 3.17 and 3.18, respectively:

$$a_{iJ} = \frac{1}{|J'_i|} \sum_{j \in J'_i} a_{ij}, \tag{3.21}$$

$$a_{Ij} = \frac{1}{|I'_j|} \sum_{i \in I'_j} a_{ij}, \tag{3.22}$$

$$a_{IJ} = \frac{1}{\upsilon_{IJ}} \sum_{i \in I'_j,\, j \in J'_i} a_{ij}, \tag{3.23}$$

$$r(a_{ij}) = \begin{cases} a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}, & \text{if } a_{ij} \text{ is non-missing} \\ 0, & \text{otherwise,} \end{cases} \tag{3.24}$$

$$H(I, J) = \frac{1}{\upsilon_{IJ}} \sum_{i \in I'_j,\, j \in J'_i} |r(a_{ij})| \,. \tag{3.25}$$

FLOC seeks the minimization of the average residue:

$$H(I, J)_{avg} = \frac{1}{K} \sum_{k=1}^{K} H(I, J)_k. \tag{3.26}$$

FLOC is able to find K biclusters simultaneously without random replacement. It starts from a random set of biclusters and then iteratively tries to move a row or a column from one bicluster to another in order to minimize the average residue. Besides that, FLOC allows simultaneous overlapping of rows and columns and is also able to deal with missing values.

Although all of these aspects can be regarded as advantages over the original model introduced by Cheng and Church, the quality of the biclustering results returned

by FLOC depends on the initial biclusters chosen, since they are chosen randomly. Therefore, the same authors propose a deterministic approach [84] where the goal is to find δ-pClusters. Given a submatrix (I, J) they consider each 2×2 submatrix $M = (I_{i_1 i_2}, J_{j_1 j_2})$ where $i_1, i_2 \in I$ and $j_1, j_2 \in J$. They also define the $pscore(M)$ as:

$$pscore(M) = \left|(a_{i_1 j_1} - a_{i_1 j_2}) - (a_{i_2 j_1} - a_{i_2 j_2})\right|. \tag{3.27}$$

A submatrix (I, J) is considered a *δ-pCluster* if for any 2×2 submatrix $M \subset (I, J)$, $pscore(M) < \delta$.

Another related approach, proposed by Pei et al. [65] is the Maximal Pattern-based Clustering (MaPle) method that relies on the property that if (I, J) is a δ-pCluster, then every submatrix (I', J') with $I' \subseteq I$ and $J' \subseteq J$ is also a δ-pCluster. Instead of computing all δ-pClusters, MaPle only enumerates all maximal δ-pClusters. A δ-pCluster (I,J) is maximal if there is no other δ-pCluster (O, P) such that $I \subset O$ and $J \subset P$.

MaPle uses the monotonicity of δ-pClusters to prune many possible combinations, since there can be a huge number of row and column combinations to consider. For a column combination J if there does not exist a set of rows I such that (I, J) is a δ-pCluster, then any superset of J is not considered .

MaPle also uses other pruning techniques, for instance, when analysing a δ-pCluster, it collects all the rows and columns that may be added to expand the current bicluster. If these candidates, together with I and J form a submatrix of a δ-pCluster that has been already found, then the search for any superset of J stops.

The methods that enumerate all biclusters, like MaPle, have mainly two advantages: they guarantee the completeness of the results and they do not miss any overlapping biclusters. However, those advantages, can also be seen as disadvantages, since such enumeration methods can be computationally expensive in large matrices.

There are also other methods in the literature which aim to find coherent biclusters. The Interrelated Two-Clustering (ITWC) [82] is a method that combines the results of

one-way clustering, on both dimensions of the data matrix, in order to find biclusters. The Double Conjugated Clustering (DCC) [14] is a method that is also based in a two-way clustering approach that allows the use of any clustering algorithm. A detailed description of these methods can be found in [53].

The biclustering methods presented so far are based on additive or multiplicative models, which do not evaluate the interaction between biclusters, i.e., the value of a certain element a_{ij} as being the sum of the contributions of the different biclusters to which the row i and the column j belong. Therefore, Lazzeroni and Owen [47] presented the plaid model defined as:

$$a_{ij} = \sum_{k=0}^{K} \theta_{ij_k} \rho_{i_k} \kappa_{j_k}, \tag{3.28}$$

where each value a_{ij} is obtained through the sum of the contribution of different biclusters (also called layers). In this model, θ_{ij_k} represents the contribution of the bicluster k to the element in the row i and the column j. In its turn, ρ_{i_k} and κ_{j_k} are binary values that indicate, respectively, the membership of the row i and the column j in bicluster k.

As Madeira and Oliveira [53] states, the plaid model described in the equation 3.28 can be seen as generalization of the additive model in the equation 3.11, since every element a_{ij} is the sum of additive models. Each bicluster $(I, J)_k$ has a contribution to the value of a_{ij} if $i \in I$ and $j \in J$.

The method proposed by Lazzeroni and Owen seeks to minimize the following merit function:

$$\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{m} \left(a_{ij} - \theta_{ij_0} - \sum_{k=1}^{K} \theta_{ij_k} \rho_{i_k} \kappa_{j_k} \right)^2, \tag{3.29}$$

where θ_{ij_0} represents the possible existence of a single bicluster that covers the whole data matrix.

The variable θ_{ij_k} can represent μ_k, $\mu_k + \alpha_{i_k}$, $\mu_k + \beta_{j_k}$, or $\mu_k + \alpha_{i_k} + \beta_{j_k}$. Thus, the plaid model is able to identify a set of constant biclusters, biclusters with constant

rows, biclusters with constant columns and biclusters with coherent values (assuming the additive model in the equation 3.11).

Madeira and Oliveira [53] also suggest a modification of the original plaid model. They consider that the value of an element a_{ij} can also be given by the product of the contributions of the different biclusters to which row i and column j belong. Similarly as the equation 3.28, the value a_{ij} is given by:

$$a_{ij} = \prod_{k=0}^{K} \theta_{ij_k} \rho_{i_k} \kappa_{j_k}. \tag{3.30}$$

The variable θ_{ij_k} is now used to represent either μ_k, $\mu_k \times \alpha_{i_k}$, $\mu_k \times \beta_{j_k}$, or $\mu_k \times \alpha_{i_k} \times \beta_{j_k}$. In its general case θ_{ij_k} is now defined by the multiplicative model in the equation 3.12.

3.4.4 Biclusters with Coherent Evolutions

Ben-Dor et al. [9] presented a method where a bicluster is defined as an order-preserving submatrix (OPSM). A submatrix (I, J) is a *order-preserving submatrix* if there is a permutation π of the set of columns J such that for each row $i \in I$ and each index $1 \leq m \leq |J|$ within the permutation $\pi(J)$ the following inequality holds:

$$a_{i\pi(J)[m]} < a_{i\pi(J)[m+1]}. \tag{3.31}$$

This is the same as saying that there is a permutation $\pi(J)$ under which the sequence of values in every row is strictly increasing.

The cluster model is defined by (J, π) and its *support* is a set of rows I that respect the inequality 3.31. In the application example of gene expression data, an OPSM is defined by a subset of genes where the expression values of each gene is strictly increasing across a permutation of a subset of conditions. This method relies on a greedy bottom-up approach in order to find the model with the highest statistical significance support. Initially it starts with small models and then it iteratively extends the best of those models.

A similar approach was presented by Liu and Wang [49]. They define a bicluster as an Order Preserving Cluster (OP-Cluster). In this method the assessment of statistical significance is discarded. Instead, the goal is to return all maximal submatrices that cover at least a certain number of rows and columns.

Murali and Kasif [60] introduced a method that aims to find conserved gene expression motifs (xMOTIFs). An xMOTIF is represented by a subset of genes (rows) where each one is in the same *state* under a subset of conditions (columns). Each state corresponds to a certain range of expression values. The goal of this method is to find the largest xMOTIF, i.e., the bicluster that contains the maximum number of conserved genes (rows). Therefore, the size of the subset of rows belonging to a certain bicluster is the merit function used to assess the quality of it.

3.5 Biclustering Validation Measures

As in cluster analysis, validation is an important issue in biclustering, since it is also considered as unsupervised learning.

Comparing biclustering methods is not a trivial task. Some methods may perform well on a certain dataset and poorly on others, since not all of them aim to discover the same type of biclusters, with the same structure. Besides that, different heuristic approaches and different merit functions are used to find and evaluate biclusters. The comparison between biclustering methods relies mainly on assessing their accuracy of recovering implanted biclusters in synthetic data. In this section we present measures to assess that accuracy.

However, in real data, most of the times, we do not have any information regarding the biclusters in it. In this section we also present approaches to quantify the meaning of the extracted biclusters. For instance, in the case of gene expression data, knowing if a group of genes, i.e., rows in a bicluster, are related is an important issue.

In clustering, both internal and external measures are used for validation. However, in biclustering, internal validation measures are seldom used [67] because it is not clear how to extend the notions of cohesion and separation to the biclustering context (overlaps and bi-dimensionality). Hence, we present two types of external validation measures: the non-biological measures and the biological measures [71].

Before presenting those measures we need to introduce some notation first. A biclustering result with K biclusters is defined as:

$$M = \{B_1, B_2, \ldots, B_K\}, \tag{3.32}$$

where B_k is the kth bicluster defined as (I_k, J_k).

Also, let M_1 and M_2 be biclustering results which consist of K_1 and K_2 biclusters, respectively. Then, each biclustering result is given by:

$$M_z = \left\{B_1^{(z)}, B_2^{(z)}, \ldots, B_{K_z}^{(z)}\right\}, \ z = 1, 2. \tag{3.33}$$

3.5.1 Non-biological Measures

The non-biological measures are used to compare a biclustering result with previous knowledge of biclusters in the data. They can also be used to compare biclusters from two different biclustering methods.

Jaccard Prelić et al. [67] proposed a measure based on the Jaccard index [36]. The Jaccard index between two biclusters B_1 and B_2 is defined as:

$$Jacc(B_1, B_2) = \frac{|I_1 \cap I_2|}{|I_1 \cup I_2|}. \tag{3.34}$$

Prelić The Prelić index [67] between two sets of biclusters M_1 and M_2 is given by:

$$Prelic(M_1, M_2) = \frac{1}{K_1} \sum_{i=1}^{K_1} \max_j \ Jacc(B_i^{(1)}, B_j^{(2)}). \tag{3.35}$$

Generally, one of the sets represents the implanted biclusters. Let M_1 be the set of implanted biclusters and M_2 the output of a biclustering method. The *average module recovery* defined by $Prelic(M_1, M_2)$ quantifies how well each of the implanted biclusters was recovered by the biclustering method. In its turn, the *average bicluster relevance* defined by $Prelic(M_2, M_1)$ quantifies how well the generated biclusters correspond to the implanted biclusters. Since the Prelić index is based on the Jaccard index only the objects (rows) are considered.

Liu and Wang [50] proposed a measure which compares two sets of biclusters considering both objects (rows) and attributes (columns):

$$LW(M_1, M_2) = \frac{1}{K_1} \sum_{i=1}^{K_1} \max_j \frac{\left|I_i^{(1)} \cap I_j^{(2)}\right| + \left|J_i^{(1)} \cap J_j^{(2)}\right|}{\left|I_i^{(1)} \cup I_j^{(2)}\right| + \left|J_i^{(1)} \cup J_j^{(2)}\right|}. \tag{3.36}$$

Turner et al. [83] adapted the F-measure to biclustering based on the definition of *sensitivity* and *specificity* adapted to biclustering by Getz et al. [27]:

$$sensitivity(B_1, B_2) = \frac{|I_1 \cap I_2|}{|I_2|} \times \frac{|J_1 \cap J_2|}{|J_2|}, \tag{3.37}$$

$$specificity(B_1, B_2) = \frac{|I_1 \cap I_2|}{|I_1|} \times \frac{|J_1 \cap J_2|}{|J_1|}, \tag{3.38}$$

$$F_1(B_1, B_2) = \frac{2 \times |I_1 \cap I_2| \times |J_1 \cap J_2|}{|I_1| \times |J_1| + |I_2| \times |J_2|}. \tag{3.39}$$

Assuming that B_1 is one of the implanted biclusters, $sensitivity(B_1, B_2)$ is the same as the average bicluster relevance, i.e., the proportion of elements in B_2 that are also on the implanted bicluster B_1. On the other hand, $specificity(B_1, B_2)$ is the same as the average module recover, i.e., the proportion of elements in the implanted bicluster B_1 that has been retrieved in B_2. The F-measure is the harmonic mean of the sensitivity and specificity.

Based on this adaptation of the F-measure, Santamaria et al. [71] proposed a measure to quantify the overall matching between two sets of biclusters:

$$Santamaria(M_1, M_2) = \frac{1}{K_1} \sum_{i=1}^{K_1} \max_j F_1(B_i^{(1)}, B_j^{(2)}). \tag{3.40}$$

The values returned by all of the measures that we have described lie in the range [0,1], where the value 1 indicates that two biclusters, or two sets of biclusters, are identical.

3.5.2 Biological measures

Biological validation is used to determine the biological relevance of the genes in a certain bicluster. In order to do that we rely on external biological knowledge, such as gene annotations from Gene Ontology (GO) [5]. GO is a collection of three, structured, controlled vocabularies of defined terms, called ontologies, which describe gene products in terms of their associated biological processes, cellular components and molecular functions in a species-independent manner.

To determine the relevance of a given bicluster, first, the genes in it are mapped to the functional categories defined in annotated databases, in this case, the GO. Thus, for a given set of genes of size n, the goal is to determine if there is a GO Term that is more represented than what it would be by chance. This technique is known as GO Term enrichment. The *significance* of a specific GO Term is calculated using a hypergeometric test. The hypergeometric test uses the hypergeometric distribution to calculate the statistical significance of having at least k genes from a bicluster with n genes by chance in a biological process containing K genes from a total size of N genes. The probability mass function of a random variable X following the hypergeometric distribution is given by:

$$P(X = k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}. \tag{3.41}$$

We can also use other biological knowledge besides GO. For instance, we can use the Kyoto Encyclopedia of Genes and Genomes (KEGG) database [42] to calculate, similarly as in GO Term enrichment, the *KEGG Pathway enrichment* of the genes in a bicluster.

However, these validation measures also have disadvantages: biological knowledge is not complete [71]. Thus, interesting biclusters can be missed. Another problem, related with GO, is that GO Terms are organized in levels, in a hierarchical structure. Relying on this structure, bigger biclusters are more likely to be enriched for more generic GO Terms, situated in the higher levels [66]. This means that the biclutering methods that return bigger biclusters are likely to have better scores than the ones returning smaller biclusters. Rhee et al. [97] presented a more complete review of the use and misuse of the gene ontology annotations.

Chapter 4

Domain Oriented Biclustering Validation

The Micro B3 project [44] investigated global marine microbial biodiversity and their functional capabilities on a single orchestrated OSD event. In the section 4.1 we present this project in more detail along with a brief exploratory data analysis of the metadata and metagenomic datasets generated from the 2014 OSD initiative.

One of the goals of our exploratory data analysis was to find geographical niches of biologically interesting groups. In order to find those groups we applied biclustering methods to the metagenomic dataset. Since the biclustering methods generated thousands of biclusters, we needed a method to evaluate each one. More precisely, we needed a biclustering validation method where the relevance of a bicluster was measured across two dimensions: the biological and the geographical dimension. Having this case study as a motivation, in the section 4.2 we present this validation task in a more general framework, where determining the biological and the geographical relevance of a bicluster is the same as determining the relevance of the rows and the relevance of the columns of any bicluster. In the section 4.3 we then propose a novel methodology that allows us to evaluate a bicluster through the relevance of the rows and the relevance of the columns belonging to it. In our methodology the concept

of what is relevant is defined by the user, since it depends on the exploratory data analysis at hand.

4.1 Case Study: Presentation

Microbes are found everywhere in the ocean and account for more than 90% of the total oceanic biomass. Microbial activity is a fundamental component of ocean's biogeochemical cycles, and are responsible for the vast majority of primary production in marine waters, the basis of the marine food web [74].

Over the past decades scientists greatly invested in achieving an understanding of the vast diversity and functions of marine microorganisms. Nevertheless, advances in their research were severed limited by the technology available. The recent advances in sequencing and computing technologies and the drastic decrease in sequencing costs, opened a new era of high-throughput metagenomic technologies, enabling the study of the complexity of marine microorganisms at a global scale. Indeed, metagenomics allows the study of microbial communities diversity and functional capabilities by analysing their genetic material recovered directly from environmental samples [28]. Several massive global scale projects are currently ongoing to analyze the World's microbiomes, like the Earth Microbiome Project [29], the Tara Oceans project [98], the Aerobiology over Antarctica project [62] and the Micro B3 project [44]. This last project investigated global marine microbial biodiversity and their functional capabilities on a single orchestrated OSD event. On June 21st 2014, scientists collected a total of 155 samples around the world for 16S/18S rRNA amplicon data to study microbial diversity, and 150 samples for metagenomes to evaluate microbial functions together with several environmental metadata. Standardized procedures for laboratory work and data processing via the Micro B3 Information System (Micro B3-IS), assured a high level of consistency and data interoperability [44]. OSD sequence and environmental contextualized data were made publicly available in the International

Nucleotide Sequence Database and at PANGAEA (archive for georeferenced data from earth and environmental science).

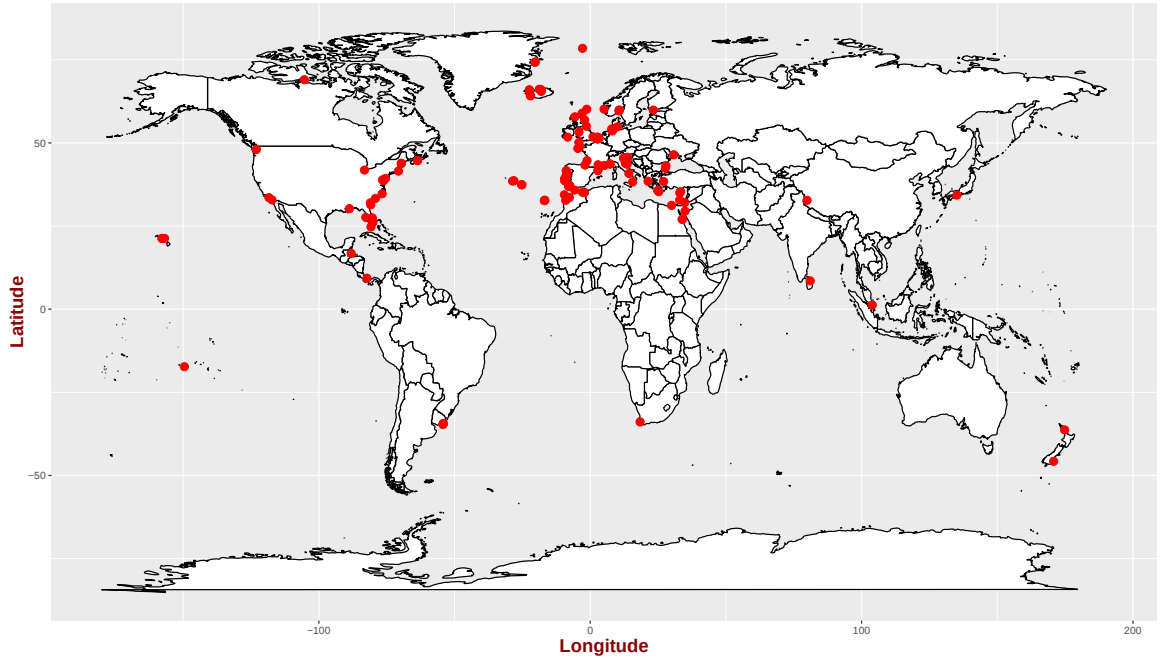


Figure 4.1: Samples distribution.

In this study, we used the metadata and metagenomic data generated from the 2014 OSD initiative. Metagenomic data were analyzed based on the European Bioinformatics Institute (EMBL-EBI) metagenomic bioinformatics pipeline [59]. This pipeline identifies rRNA sequences, using rRNASelector, and performs taxonomic analysis upon 16S rRNAs using Qiime. The remaining reads are submitted for functional analysis of predicted protein coding sequences using the database of protein families, InterPro sequence analysis resource. InterPro uses diagnostic models to classify sequences into families and to predict the presence of functionally important domains. By utilizing this resource, the service offers a powerful and sophisticated alternative to BLAST-based functional metagenomic analyses [59]. We used the normalized InterPro read counts table generate by EMBL-EBI, constituted by a total of 15008 different protein families with a specific accession number (IPR$xxxxxx$) across 150 sites. In Figure 4.1 we can observe that the samples were collected from marine sites spread all

over the world, even though there is a larger number of samples collected in Europe and North America in relation to other continents.

Due to the high dimensionality of the data an overall exploratory data analysis is unfeasible, thus we focused our analysis on the proteins/enzymes involved on the major nitrogen cycle pathways particularly on the N-fixation, nitrification, denitrification and dissimilatory nitrate reduction to ammonia (DNRA). Nitrogen is recognized as an essential element for the functionality and sustainability of ecosystems since it is fundamental to maintain the microbial metabolism that sustains global biogeochemical cycles (e.g. Francis et al. [26]). Thus its transformation pathways mediated by microbial functions are essential to maintain ocean's ability to sustain life.

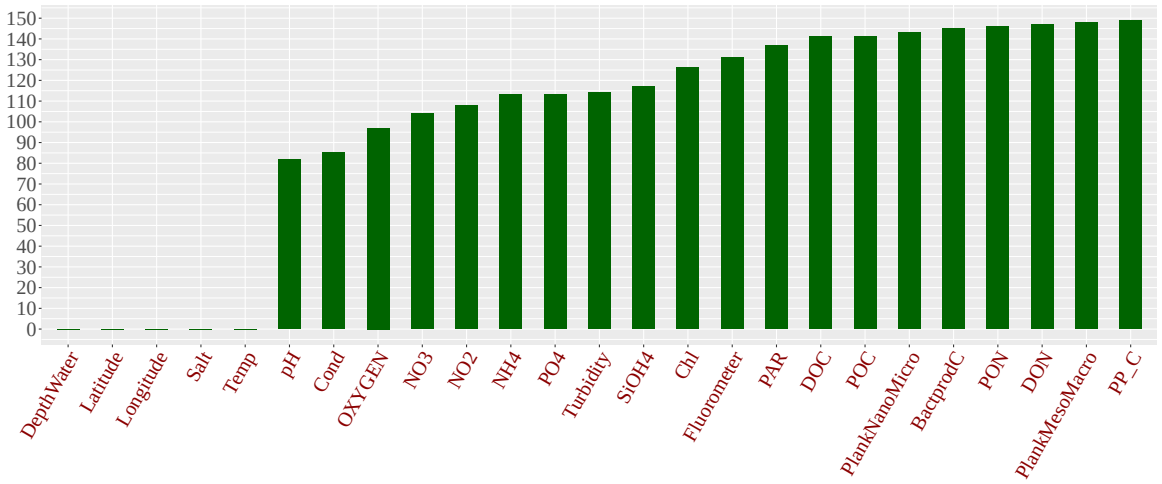


Figure 4.2: Number of missing values for each variable.

The marine microbial data generated during the OSD initiative was environmentally contextualized by analyzing several environmental parameters that characterized the 150 marine sites. This metadata dataset is formed by 25 variables describing each site: *Latitude, Longitude, DepthWater, Temp, Cond, Salt, PAR, Turbidity, pH, OXYGEN, NO3, NO2, NH4, PO4, SiOH4, POC, PON, DOC, DON, Fluorometer, Chl, PlankNanoMicro, PlankMesoMacro, PP_C,* and *BactprodC.*

We started a brief exploratory analysis of the metadata dataset by examining the missing values. The number of missing values for each variable is depicted in Figure

4.2. Due to the high number of missing values, for the majority of the variables we just used the *Latitude*, *Longitude*, *DepthWater*, Temperature (*Temp*) and Salinity (*Salt*), which were mandatory environmental variables with no missing values over the whole dataset. Figure 4.3 shows the distribution of these variables after standardization to allow simultaneous presentation on a single graph. From this figure we can observe that *DepthWater* presents severe outliers, although, most of the samples were collected in low depths. In its turn, the observations of *Temp* are evenly split at the median, which suggest a symmetric distribution. There are also some outliers in *Temp*, which correspond to samples collected in cold waters. The *Salt* variable presents a high range of variation and has several outliers. Table 4.1 provides a summarization of these environmental variables according to some statistical measures.

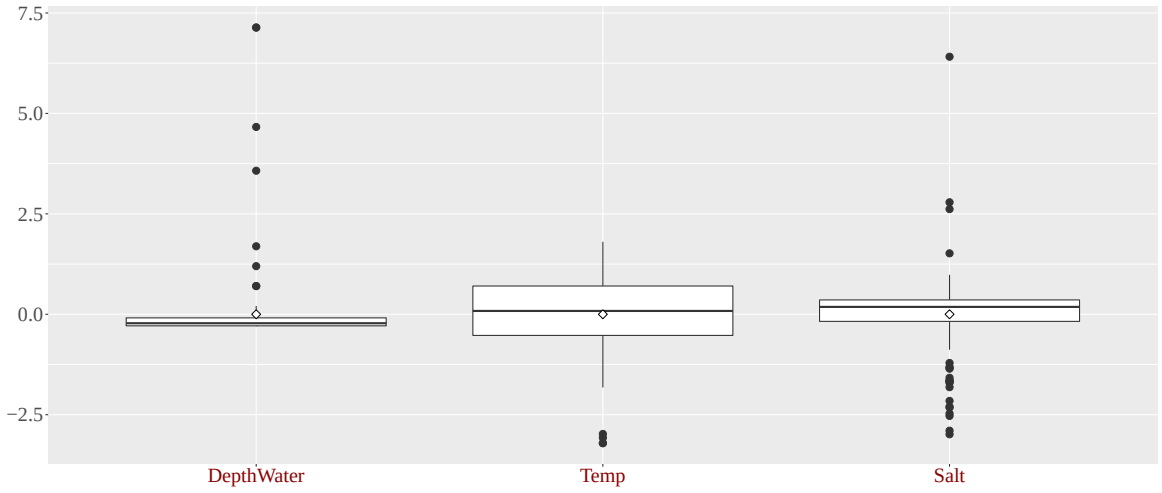


Figure 4.3: Distributions of environmental variables.

In order to evaluate the significant correlations between these environmental variables we used the *Spearman's rank correlation coefficient* at the significance level of 0.05. Hence, in Figure 4.4 we depict only the correlations that are statistically significant. We can observe from Figure 4.4 that there is a significant correlation between *DepthWater* and *Temp*. The mean temperature of the samples where *DepthWater* is > 2 (corresponding to the top 25% of the highest depths) is 15.58, which is lower than the global mean temperature of 19.46 and lower than the mean temperature of 20.42 corresponding to the samples where *DepthWater* is ≤ 2. This correlation suggests

that the samples collected in higher depths have an overall lower temperature from the ones collected on the surface.

	Temp	Salt	DepthWater
Min	-1.61	0.14	0.00
Max	31.30	100.00	75.00
Median	20.00	33.83	0.65
Mean	19.46	31.88	2.90
Variance	43.01	112.88	102.00
1st Qu.	16.00	30.00	0.00
3rd Qu.	24.07	35.68	2.00

Table 4.1: Summarization of environmental variables.

Relating the geographical information of each sample with the microbial functions distribution along the 150 OSD sites (metagenomic data) can be an important step to identify geographical niches of certain microbial functional capabilities. Thus, besides the previous analysis, we tried to find similar groups of IPRs in the metagenomic dataset and we also tried to relate those groups with the geographic information and the environmental characteristics mentioned above, i.e., location, temperature, salt and depth of the water.

In order to find similar groups of IPRs we could have applied some clustering methods described in Chapter 2. However, we would have to deal with a set of problems related with the high dimensionality of the data known as the *curse of dimensionality* [2,8,45]. The main challenge for clustering, in this case study, is how to deal with the presence of irrelevant attributes, i.e., there are several attributes but not all of them are relevant for each cluster. Different subsets of attributes are relevant for different clusters. Kriegel et al. [45] call this phenomenon the *local feature relevance*. Techniques of dimensionality reduction, such as the PCA are not suitable for this particular problem because they derive only a subset of correlated attributes out of thousands of possible subsets where we can also find clusters.

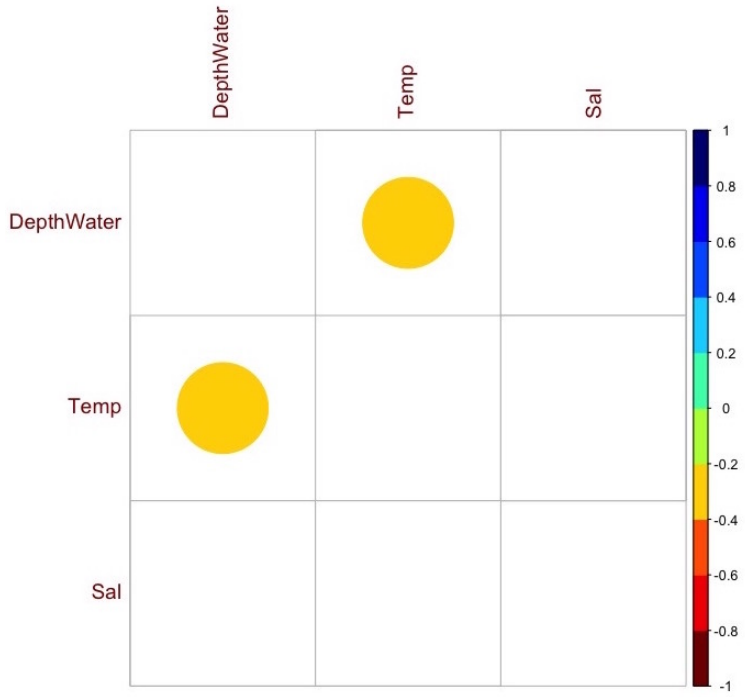


Figure 4.4: Significant correlations

In this particular case study, we wanted to find subsets of IPRs that express similarly under a subset of samples, since trying to find a group of IPRs that express similarly in all samples is not realistic from a biological point of view. Hence, we applied some of the biclustering methods described in Chapter 3 in order to find those groups. This step is described in the section 5.1.

Although biclustering is an approach that allowed us to deal with the local feature relevance phenomenon we still had the validation problem, i.e., we needed to evaluate the generated biclusters. Besides determining the biological significance of a bicluster we also wanted to determine its significance from a geographical point of view, since we were interested to find geographical confined subsets of IPRs with similar expression patterns. More generally, we wanted to evaluate biclusters according to their rows and columns, without prior information about the existing biclusters in the data. In the next section we present this problem in a more general way and discuss it in the context of what has already been presented in the literature.

4.2 Problem Statement

The problem of clustering validation has been extensively discussed in the literature [2]. However, there has not been the same amount of work on comparison and validation of biclustering results [67]. Although biclustering was first proposed by Hartigan [35] in 1972, it only found an application in 2000 [18]. Based on the work of Cheng and Church, biclustering became an important technique in the field of bioinformatics, more precisely in gene expression analysis. As a consequence, several biclustering methods and validation measures were proposed to find and evaluate biclusters in biological data. In the section 3.5 we have described two types of measures: biological measures and non-biological measures. The biological measures rely on external sources of biological information in order to assess the relevance of a certain bicluster. On the other hand, the non-biological measures are used to compare a biclustering result with implanted biclusters in synthetic data. None of those measures rely solely on the information intrinsic to the data. In fact, internal measures, as described in clustering, are seldom used in biclustering, mainly because the concepts of cohesion and separation are non trivial to adapt to biclustering due to the bi-dimensionality and overlap of the biclusters.

Most of the times, in real data, we do not have any information about the biclusters in it. Information about the types of biclusters and noise in the data is crucial in order to choose the proper biclustering method. Besides the choice of the method, the quality of a biclustering result is also related with choosing the proper parameters for each biclustering method. The Gene Set Enrichment is widely used to assess the quality of a biclustering result, since it does not require prior information about the data. It evaluates the relevance of the genes in the biclusters based on the biological information available in public databases, such as the GO. However, the biological knowledge in GO is not complete. Besides other disadvantages presented in the section 3.5, the evaluation provided by the Gene Set Enrichment is only related with genes, i.e., the conditions (columns) are ignored.

The measures that we have presented so far allow us to compare a biclustering results with known implanted biclusters, which is not too useful in real data, and evaluate the biological significance of a group of genes (rows) within a bicluster. Hence, the main problem is how to assess the relevance of a biclustering result w.r.t. the rows and the columns in cases where we do not have prior information about the biclusters in the data and we cannot rely on GO.

4.3 Methodology

The concept of what constitutes a *relevant bicluster* is too vague. A bicluster may be relevant under some context and irrelevant on another. When there is no ground truth about the data, the evaluation of biclustering results may depend on interpretations of human experts [67].

Biclustering can be a fundamental technique in exploratory data analysis, but it cannot be seen as a way of discovering an absolute truth of the data objects, i.e., there can be several interesting biclusters. Hence, without the presence of a ground truth, the notion of what is a relevant bicluster is always related with the application domain and with the goal of the exploratory data analysis at hand.

Therefore, we propose a biclustering validation method where the definition of what constitutes a relevant bicluster relies on the nature of the data and on the expertise of human experts. The method also takes into account the rows and the columns of each bicluster.

We evaluate a bicluster through a *score*, which consists in a weighted arithmetic mean between the values returned by two functions. One of those functions is used to determine the relevance of the rows, we will call it *score of the rows*, whereas the other one is used to determine the relevance of the columns (*score of the columns*). Let $B = (I, J)$ be a bicluster. Then, its score is given by:

$$Score(B) = w_1 \cdot f_{rows}(I) + w_2 \cdot f_{cols}(J), \tag{4.1}$$

where $f_{rows}(I)$ is the score of the rows, $f_{cols}(J)$ is the score of the columns, and w_1 and w_2 are weights assigned by the user, such that $w_1 + w_2 = 1$. Those weights allow us to quantify the importance of each score for the end-user. In the case of $w_1 = w_2$ the final score is just the arithmetic mean between the score of the rows and the score of the columns.

In order to calculate the score of the rows belonging to a certain bicluster we use a set of indexes where each index quantifies some property related with the rows. Those indexes are defined by the user and they vary according to the nature of the data and with the application domain. Let $X = \{x_1, x_2, \ldots, x_Z\}$ be a set of Z indexes w.r.t. the rows. The score of the rows is then given by a linear combination of the rows indexes:

$$f_{rows}(I) = \alpha_1 \cdot x_1(I) + \alpha_2 \cdot x_2(I) + \cdots + \alpha_Z \cdot x_Z(I), \tag{4.2}$$

where α_l is the weight assigned by the user to the l^{th} row index, such that $\sum_{l=1}^{Z} \alpha_l = 1$.

The score of the columns is defined similarly as the score of the rows. Let $Y = \{y_1, y_2, \ldots, y_U\}$ be a set of U indexes w.r.t. the columns. The score of the columns is then given by a linear combination of the column indexes:

$$f_{cols}(J) = \beta_1 \cdot y_1(J) + \beta_2 \cdot y_2(J) + \cdots + \beta_U \cdot y_U(J), \tag{4.3}$$

where β_l is the weight assigned by the user to the l^{th} column index, such that $\sum_{l=1}^{U} \beta_l = 1$.

Although we already have described how to calculate the score of a bicluster, we still did not present how to evaluate a group of biclusters (biclustering result). A trivial solution would be to iterate through the biclusters and calculate the score for each one. However, that approach raises a problem related with the different scales of the values of the indexes. Suppose that we have an index whose values are between 1 and 100, and another one whose values are between 0 and 1. If we calculate a score based on those indexes, most of the times, the index whose values are between 1 and 100 will have more importance, despite the weights assigned to each index.

Therefore, in order to evaluate a biclustering result we start by calculating the values of the indexes of the rows and the values of the indexes of the columns for each bicluster. After that we normalize the values of each index, so the values in all of them lie on the same scale (note that the function used to normalize the indexes cannot change the distribution of the values). In the next step we calculate the score of the rows and the score of the columns for each bicluster using the equation 4.2 and the equation 4.3, respectively. Finally, we calculate the final score for each bicluster using the equation 4.1. The algorithm of the presented method is described in Algorithm 1.

Algorithm 1 Evaluation of a biclustering result

input : $B = \{B_1, B_2, \ldots, B_K\}$ the biclustering result

input : $X = \{x_1, x_2, \ldots, x_Z\}$ the set of rows indexes

input : $Y = \{y_1, y_2, \ldots, y_U\}$ the set of columns indexes

input : w_1 the weight assigned to the score of the rows

input : w_2 the weight assigned to the score of the columns

input : $\alpha_1, \alpha_2, \ldots, \alpha_Z$ the weights assigned for each row index

input : $\beta_1, \beta_2, \ldots, \beta_U$ the weights assigned for each column index

output: V_{score}, the vector with the scores for each bicluster

1: let M_{rows} be a new $K \times Z$ matrix
2: let M_{cols} be a new $K \times U$ matrix
3: let $V_{rows}[1, \ldots, K]$, $V_{cols}[1, \ldots, K]$ and $V_{score}[1, \ldots, K]$ be new arrays
4: **for** $B_i = (I_i, J_i) \in B$ **do**
5: **for** $x_l \in X$ **do**
6: $M_{rows}[i, l] \leftarrow x_l(I_i)$
7: **for** $y_l \in Y$ **do**
8: $M_{cols}[i, l] \leftarrow y_l(J_i)$
9: **for** $l \leftarrow 1$ **to** Z **do**
10: `normalize`$(M_{rows}[:, l])$
11: **for** $l \leftarrow 1$ **to** U **do**
12: `normalize`$(M_{cols}[:, l])$
13: **for** $i \leftarrow 1$ **to** K **do**
14: $V_{rows}[i] \leftarrow \alpha_1 \cdot M_{rows}[i, 1] + \ldots + \alpha_Z \cdot M_{rows}[i, Z]$
15: $V_{cols}[i] \leftarrow \alpha_1 \cdot M_{cols}[i, 1] + \ldots + \alpha_U \cdot M_{cols}[i, U]$
16: $V_{score}[i] \leftarrow w_1 \cdot V_{rows}[i] + w_2 \cdot V_{cols}[i]$
17: **return** V_{score}

Chapter 5

Case Study

In the last chapter we presented a case study where the goal was to find geographic confined subsets of IPRs with similar expression patterns. The application of biclustering techniques was suggested in order to find those groups. However, we needed a way to validate our biclustering results, since we did not have any prior information about the biclusters in the data. We also needed to evaluate the biclusters from a geographical and biological point of view. Therefore, in the section 4.3 we presented a general methodology to evaluate biclusters that takes into account the relevance of the rows and the relevance of the columns through a set of indexes defined according to the application domain.

In this chapter we present an application of the suggested methodology to our case study. In order to apply the methodology we started by generating a set of biclusters. Then, we defined the set of rows indexes and the set of columns indexes. This is described in the section 5.1.

In the section 5.2 we present ORCA, which is a web application that we developed. ORCA allows us to compute the scores of the generated biclusters relying on the set of indexes that we defined. The weights for those indexes are assigned interactively by the user.

We used ORCA to evaluate the generated biclusters. In the section 5.3 we present a biological interpretation for some of those biclusters.

5.1 Case study application

In order to apply the suggested methodology to our case study we started by generating a set of biclusters. Since we did not have any prior information about the types of biclusters in the data we used several biclustering methods and ran them several times with different parameters, leading to a large set of biclusters. Then, we defined a set of indexes to evaluate the relevance of the rows and another set of indexes to evaluate the relevance of the columns. We finally applied Algorithm 1 to evaluate our set of biclusters.

All analysis were performed using R (version 3.2.4), and the biclusters were generated with the biclust R package (version 1.2.0) [41]. The full environment can be found in Appendix A.

5.1.1 Collection

In Chapter 3 we presented several types of biclusters and several biclustering structures. We also presented some biclustering methods, which rely on different heuristic approaches and different merit functions in order to find and evaluate biclusters. Therefore, the quality of a biclustering result depends on the method and the parameters chosen. Since we did not have any information about the types of biclusters in the metagenomic dataset, we applied several biclustering methods and ran them several times with different parameters. We obtained a set of different biclustering results, which we call *collection*.

In order to generate our collection of biclusters we used the implementations of the following methods provided by the package biclust: Cheng and Church [18], Plaid

Model [47], Spectral [43] and Quest [83]. We did not use the Bimax [67] and the xMOTIF [60] methods because they require a discrete matrix and the transformation of the metagenomic data was out of the scope of our work. The parameters used for each biclustering method can be found in Appendix A.

5.1.2 Rows Indexes

We defined a set of domain-specific rows indexes in order to assess the relevance of the IPRs belonging to a bicluster (note that the rows of the generated biclusters correspond to IPRs). Since we have thousands of IPRs in the metagenomic dataset, as we mentioned in the section 4.1, we focused our analysis on the proteins/enzymes involved on the major nitrogen cycle pathways, particularly on the N-fixation, nitrification, denitrification and dissimilatory nitrate reduction to ammonia. Following the advice of domain experts, we defined four groups of *interest* that are presented in Table 5.1. In this case, a bicluster is considered relevant, from a biological point of view, if the IPRs belonging to it are somehow related with the major nitrogen cycle pathways.

In order to assess the relevance of a bicluster, most of the indexes that we defined take into account the groups presented in Table 5.1. Moreover, some of those indexes are adaptations of some of the clustering validation measures described in the section 2.5. We define those indexes below.

Let $A = (N, M)$ be the data matrix representing the metagenomic dataset, where N is the set of IPRs and M is the set of samples. Let $B = (I, J)$ be a bicluster, such that $I \subseteq N$ and $J \subseteq M$. Also let X_1, X_2, X_3 and X_4 be the sets of IPRs involved on the N-fixation, nitrification, denitrification and DNRA, respectively.

nIPRs The fraction of IPRs covered by the bicluster:

$$nIPRs(I) = \frac{|I|}{|N|}. \tag{5.1}$$

N-Fixation	**Nitrification**	**Denitrification**	**DNRA**
IPR003731	IPR003393	IPR003816	IPR003321
IPR024564	IPR006833	IPR004448	IPR005117
IPR030655	IPR006980	IPR005591	IPR006067
IPR000392	IPR024656	IPR005623	
	IPR003393	IPR010649	
	IPR012138	IPR028189	
		IPR003143	
		IPR001287	
		IPR010266	
		IPR001505	
		IPR007742	
		IPR008719	

Table 5.1: Interesting IPRs.

Precision The fraction of IPRs within the bicluster that are interesting:

$$Precision(I) = \frac{|I \cap X_1| + |I \cap X_2| + |I \cap X_3| + |I \cap X_4|}{|I|}. \tag{5.2}$$

Recall The fraction of interesting IPRs covered by the bicluster:

$$Recall(I) = \frac{|I \cap X_1| + |I \cap X_2| + |I \cap X_3| + |I \cap X_4|}{|X_1| + |X_2| + |X_3| + |X_4|}. \tag{5.3}$$

F-measure The weighted average of precision and recall. It measures the extent to which a bicluster contains only interesting IPRs and all the interesting IPRs:

$$F_1(I) = \frac{2 \cdot Precision(I) \cdot Recall(I)}{Precision(I) + Recall(I)}. \tag{5.4}$$

The next indexes are the same as the recall of each interesting group of IPRs.

N-Fixation The fraction of IPRs belonging to the N-Fixation covered by the bicluster:

$$NFixation(I) = \frac{|I \cap X_1|}{|X_1|}. \tag{5.5}$$

Nitrification The fraction of IPRs belonging to the nitrification covered by the bicluster:

$$Nitrification(I) = \frac{|I \cap X_2|}{|X_2|}. \tag{5.6}$$

Denitrification The fraction of IPRs belonging to the denitrification covered by the bicluster:

$$Denitrification(I) = \frac{|I \cap X_3|}{|X_3|}. \tag{5.7}$$

DNRA The fraction of IPRs belonging to the DNRA covered by the bicluster:

$$DNRA(I) = \frac{|I \cap X_4|}{|X_4|}. \tag{5.8}$$

5.1.3 Columns Indexes

The goal of our analysis was to find geographic niches of IPRs involved on the major nitrogen cycle pathways. Thus, we also defined a set of columns indexes in order to assess the geographical relevance of a bicluster. A bicluster is geographically relevant if the samples belonging to it are somehow close to each other in a geographical context. Hence, the columns indexes that we defined are based on the geographical location from where each sample was taken, i.e., its latitude and longitude.

For some indexes a quantification of the distance between sampling sites was required. We then computed the great circle distance using the `WGS84` ellipsoid with the R package `sp` [11, 63]. Other indexes require the country, or the continent from where each sample was taken. Therefore, in order to obtain the country and the continent for each sampling site we used the R package `ggmap` [40]. We define those indexes below.

Let $A = (N, M)$ be the data matrix representing the metagenomic dataset, where N is the set of IPRs and M is the set of samples. Let $B = (I, J)$ be a bicluster, such that $I \subseteq N$ and $J \subseteq M$. Also let $max\text{-}country(J)$ be a function that returns the number of appearances of the most represented country in the bicluster and $max\text{-}continent(J)$ be a function that returns the number of appearances of the most represented continent in the bicluster.

nSamples The fraction of samples covered by the bicluster:

$$nSamples(J) = \frac{|J|}{|M|}. \tag{5.9}$$

Countries Prevalence The fraction of the samples in the bicluster belonging to the most represented country in it:

$$CountriesPrevalence(J) = \frac{max\text{-}country(J)}{|J|}. \tag{5.10}$$

Continents Prevalence The fraction of samples in the bicluster belonging to the most represented continent in it:

$$ContinentsPrevalence(J) = \frac{max\text{-}continent(J)}{|J|}. \tag{5.11}$$

Dispersion This index quantifies the geographical dispersion of the samples belonging to the bicluster. Let $dist_{i,j}$ be the geographical distance between the sample S_i and the sample S_j, and D_i be the set of the geographical distances between S_i and the other samples defined as:

$$D_i = \{dist_{i,j} \mid \quad S_j \in J \setminus \{S_i\}\}. \tag{5.12}$$

Therefore we defined the dispersion as:

$$Dispersion(J) = \min\{median(D_i) \mid \quad \forall S_i \in J\}. \tag{5.13}$$

We could also use the arithmetic mean, but the median is less sensitive to outliers.

5.1.4 Biclustering Evaluation

Besides the set of rows indexes and the set of columns indexes, in order to evaluate the biclusters belonging to our collection, Algorithm 1 requires the weight assigned to the score of the rows, the weight assigned to the score of the columns, a set with the weights assigned for each row index, and a set with the weights assigned for each column index. The assignment of those weights is always related with the exploratory data analysis at hand. In the next section we will present the combinations of the weights that we used.

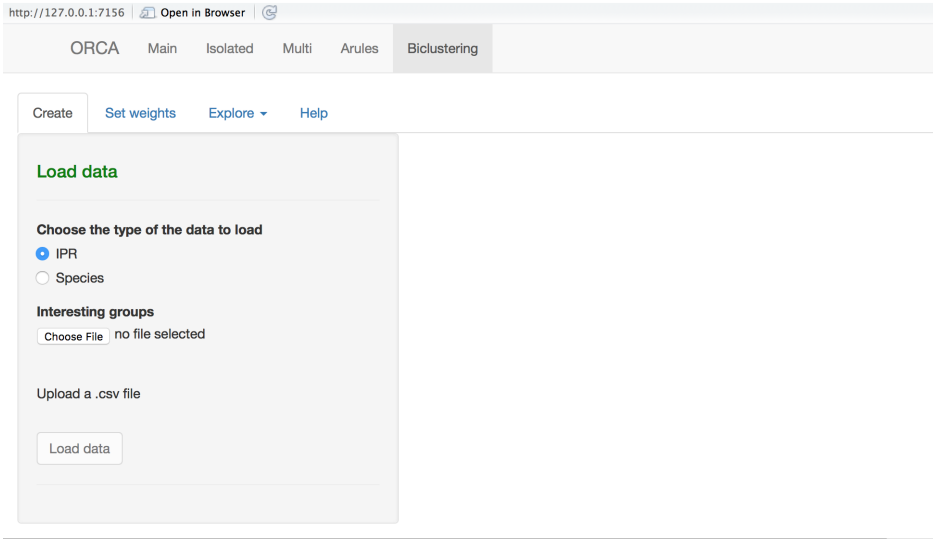


Figure 5.1: Biclustering main menu.

The algorithm also requires a normalization function, so the values of each index lie on a scale between 0 and 1 (the optimal value of 1 is assigned to the maximum value of each index). We used a unity-based normalization. Note that, contrarily to the other indexes, a low *Dispersion* value is always preferable instead of a higher one. Thus, the optimal value of 1 corresponds to its minimum value. We achieved this by computing the difference between 1 and the standard normalized value.

5.2 ORCA Application

We developed an interactive web application, known as ORCA[1], where we implemented the methodology described in the section 4.3. ORCA was developed using Shiny [17], which is a web application framework for R, and uses the preprocessed metadata and metagenomic datasets generated from the 2014 OSD initiative.

In order to increase the performance and scalability of the server, ORCA works with a previously generated collection of biclusters, since the process of computing a collection of biclusters in real time takes too long, usually days, and it is also very demanding for the server.

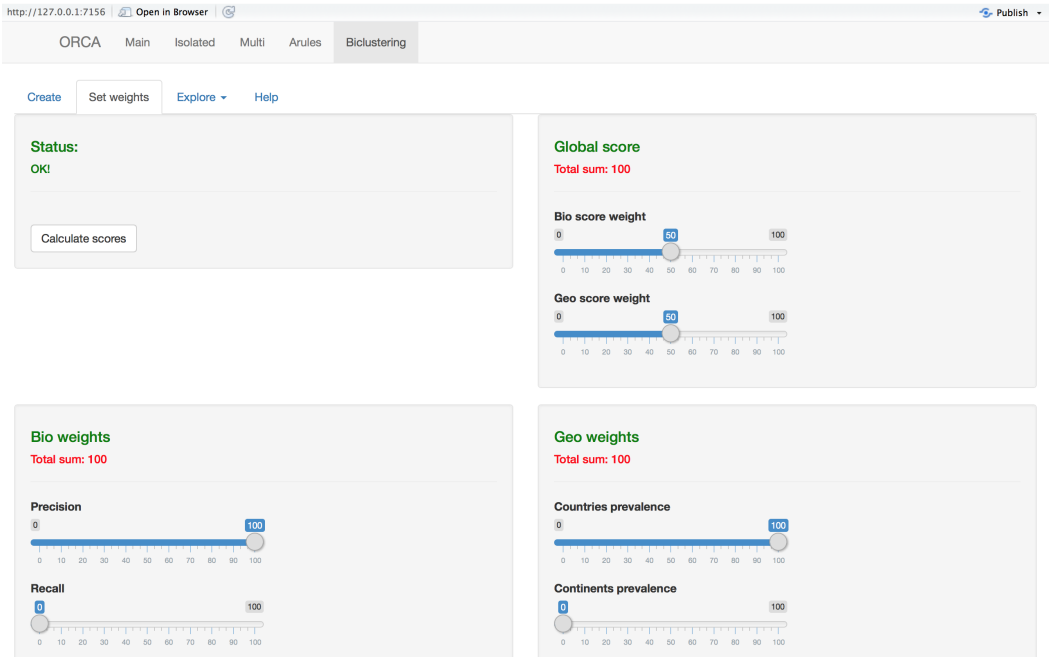


Figure 5.2: Set weights.

ORCA is an interactive application: it allows us to define the interesting groups of IPRs and it also allows us to assign the weights of the rows indexes (biological indexes), the weights of the columns indexes (geographical indexes), the weight of the biological score and the weight of the geographical score. Some of the biological indexes change

[1] avaliable at https://carlosleite.shinyapps.io/orca/

accordingly with the interesting groups of IPRs (the *nIPRs*, *Precision*, *Recall* and *F-measure* are the permanent indexes), whereas the geographical indexes are the ones presented in the subsection 5.1.3.

Since some of the biological indexes are related with the interesting groups of IPRs, before the assignment of the weights, ORCA requires that we first upload a file with the interesting IPRs. Figure 5.1 presents the menu where we can upload the file with the interesting groups. After the upload we can interactively assign the weights, as depicted in Figure 5.2. As we mentioned in the section 4.3, in order to compute the scores of each bicluster, the sum of the scores in each panel should be 100, which stands for the percentage units. The scores are computed using Algorithm 1.

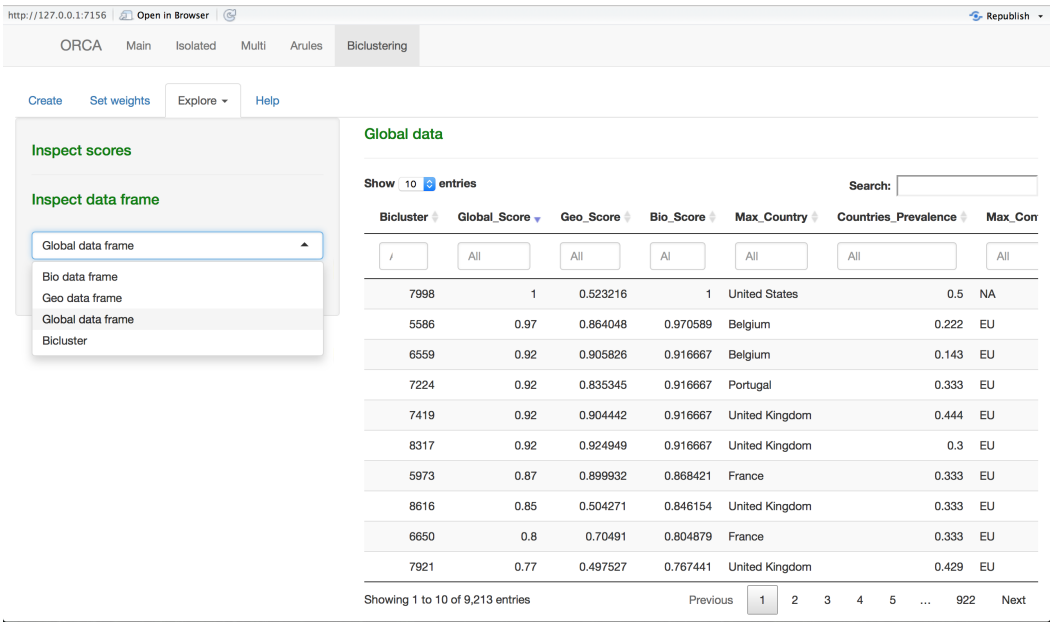


Figure 5.3: Inspect biclusters.

ORCA also provides tools to inspect the global, biological and geographical scores of each bicluster. Those scores are presented along with the indexes that were used to calculate them, as shown in Figure 5.3. If we choose the *Bicluster* option we can obtain the rows and the columns which belong to a certain bicluster.

In order to enhance the exploratory analysis of the biclusters, ORCA allows us to visualize a bicluster through a set of plots. We can obtain, as depicted in Figure 5.4, a world map with the location of the samples belonging to a certain bicluster. We can also obtain other plots, such as parallel plots, correlation plots and heatmaps.

ORCA is also prepared to identify biclusters at the taxonomic level (species identification), although our work relied only on the microbial functions (IPRs).

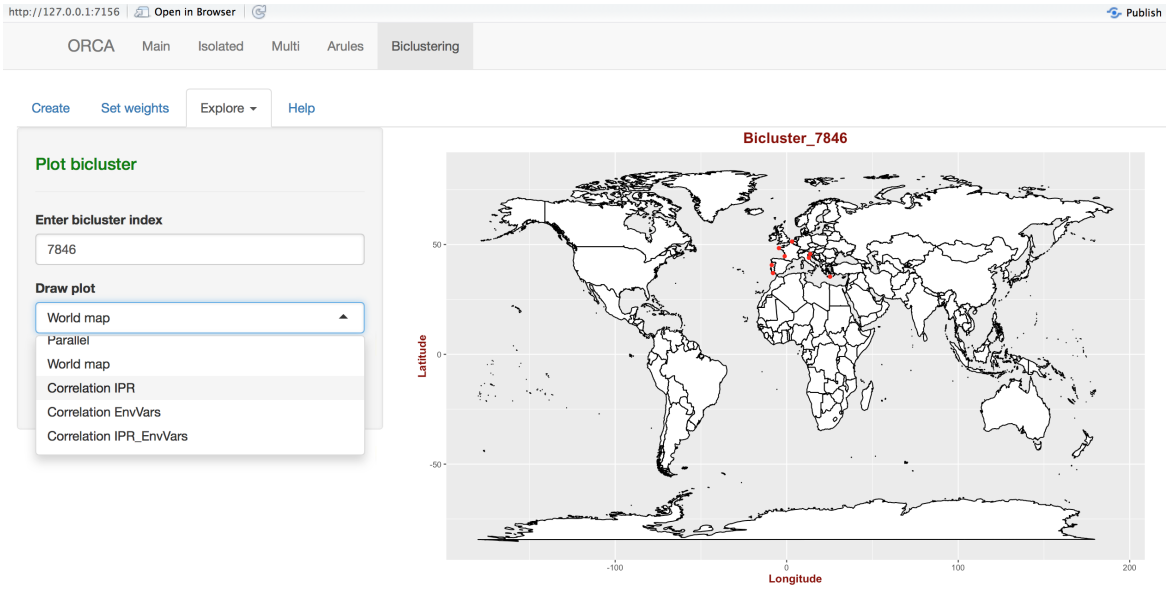


Figure 5.4: Biclustering visualization.

5.2.1 Other Features

Besides the biclustering evaluation, ORCA also provides a set of tools to explore the metadata and metagenomic datasets at function (IPRs) and taxonomic (16S rRNA gene) levels generated from the 2014 OSD initiative. We analysed the distribution and environmental controls on marine nitrogen biogeochemical functions along the OSD datasets using ORCA [54].

Figure 5.5 depicts the main menu, where we can choose what type of data we want to load. It is also possible to filter these datasets by uploading a file with the selected

IPRs or Species of interest. After loading the data we can individually analyse each IPR/Specie or the selected group of IPRs/Species through a set of plots.

`ORCA` also provides a module for frequent item set mining. It relies on the package `arules` [32], which provides an implementation of the *apriori* algorithm [4, 13] with some improvements. However, besides the minimum support, minimum confidence and the minimum length of a rule, the user also needs to provide a threshold t in order to discretize the data into two groups (1 for the values $\geq t$ and 0 for the values $< t$). The generated rules can be explored through a set of plots.

All the plots generated through `ORCA` can be downloaded. Besides those features, we also implemented some validations, since some errors could break the app. For instance, trying to filter an IPR that does not exist in our dataset causes an error. Hence, we return a personalized message every time that the user tries to do something that can cause an error.

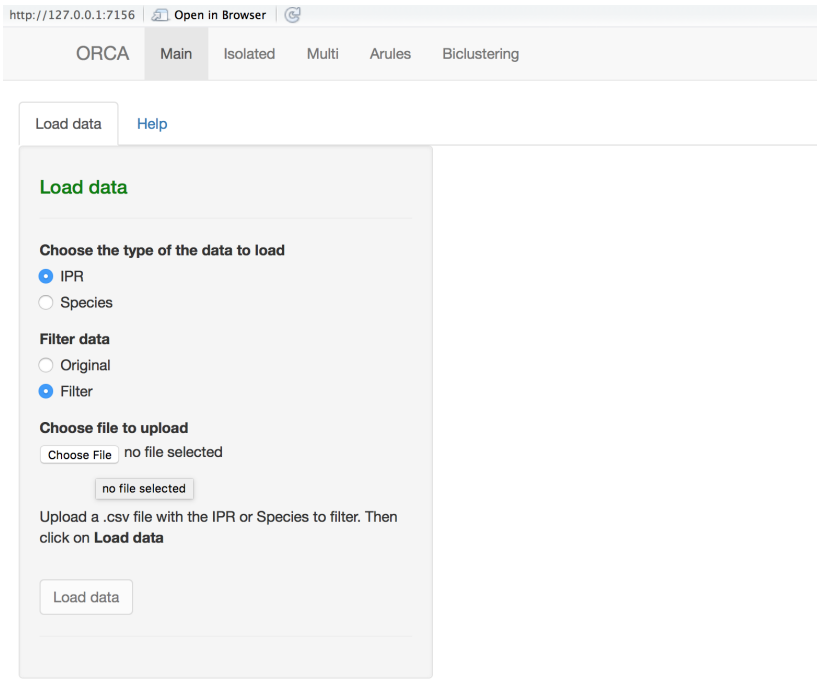


Figure 5.5: Load data.

5.3 Results

We used ORCA to evaluate our collection of biclusters. We started by uploading a file with the interesting groups of IPRs presented in Table 5.1. Then, we tried to find geographical niches of each of those groups by computing the scores of the biclusters several times. In each time we assigned different weights for each biological index, according with the specific function that we were interested to explore. On the other hand, we always assigned the same weight for the geographical indexes (100 for the *Dispersion*). We also assigned equal weights for the biological score and the geographical score. For instance, in order to find a geographical niche of IPRs belonging to the N-fixation we assigned 100 for the *N-Fixation* index and 100 for the *Dispersion* index. Thus, we computed the scores of the biclusters four times.

After computing the scores of the biclusters, in each time, we searched for biclusters with a high global score and with a low *nIPRs* value. Note that *nIPRs* is the fraction of IPRs covered by the bicluster, so the higher the values of the *nIPRs* index the higher is the recall of each group. The problem here is that a bicluster with several IPRs, i.e. with a high *nIPRs* value, is not interesting since we are searching for niches of IPRs.

ORCA allowed us to find several interesting biclusters in our collection. Next we will present two of those biclusters along with a biological interpretation of the IPRs and the samples belonging to them. The analysis that supported the biological findings can be found in Appendix B.

Bicluster 6559 According to domain experts, this bicluster is very interesting from a biological point of view. It includes a group of samples from a confined geographic area where the enzyme system responsible for biological nitrogen fixation (represented by IPR030655 and IPR000392), the process that makes nitrogen available for all living organisms, was found to be tightly related to other biological functions (IPR010241, IPR018639, IPR028090).

These observations imply that the enzyme system responsible for biological nitrogen fixation had strong functional associations with plant-type ferredoxins (IPR010241), with a prokaryotic protein with still an unknown function (IPR018639) and with prokaryotic JAB peptidases (IPR028090). Species with plant-type ferredoxins tend to be photosynthetic, suggesting that photosynthetic prokaryotes, like cyanobacteria, might be key players of Nitrogen fixation in this geographic area. The tight interconnectivity identified within the IPRs included in this bicluster suggested that the enzymes involved in Nitrogen fixation and the ones that represented the IPRs highly correlated are likely to function together in signalling systems, many of those not yet identified.

Moreover the IPRs represented in this bicluster are inversely correlated with salinity, suggesting that the operation of those tight biological functions are stimulated under lower salinity conditions (coastal marine environments, instead of open ocean environments).

Bicluster 9093 According to domain experts, this is also an interesting bicluster since it identifies a very close relationship between IPR006067, a domain of related nitrite reductases (NiRs), which catalyse the six-electron reduction reactions of nitrite to ammonia, with other biological functions (IPR002641, IPR003718, IPR003732, IPR006067, IPR010066, IPR015795 and IPR021497). This reaction comprises the second pathways of the DNRA, a widely distributed nitrogen pathway in natural habitats, with an important role in retaining bioavailable fixed nitrogen within the ecosystems by producing ammonium (NH_4^+) as a dissimilatory end product of nitrate (NO_3^-) reduction.

This bicluster allowed us to identify strong relationships between nitrite reductase in DNRA and other cell biological functions like osmotically stress regulation (peroxiredoxins, IPR002641), D-aminoacyl-tRNA deacylase DTD (IPR003732) an ezymatic system that cleaves a lethal complex, in order to rescue cell by releasing tRNA molecules, with pyruvate kinase (IPR015795) which is an enzyme that catalyzes the

final step of glycolysis and with bacterial 3TM Holins (IPR021497) a diverse group of small proteins produced that control the degradation of the host's cell wall. It is interesting to note that neither of those function interconectivities have been yet experimentally described and must be explored in future investigations.

Chapter 6

Conclusion

In this dissertation we presented the 2014 OSD event as our case study. Within this problem our aim was to find geographical confined subsets of IPRs involved on the major nitrogen cycle pathways.

Through the application of biclustering techniques we were able to find subsets of IPRs with similar expression values in a subset of samples. However, we still needed to evaluate the relevance of the generated biclusters from a geographical and biological point of view. We addressed this problem by proposing a general methodology which evaluates a bicluster considering the rows and the columns belonging to it.

In the presented methodology, the relevance of a bicluster is quantified through a score. That score consists in a weighted arithmetic mean between the values returned by two functions. One of those functions is used to determine the relevance of the rows (score of the rows), whereas the other one is used to determine the relevance of the columns (score of the columns). The overall design goal of the proposed methodology was to allow for an easy specification of the preference biases of particular domains of application. The motivation lies on the fact that different application domains may have different criteria to define what is an interesting bicluster. In this context, we aimed at providing a general schema that allowed the users to easily insert their preference biases regarding both the rows and the columns forming a bicluster.

Moreover, the methodology uses a set of weights that allows the user to interactively play with their values to easily explore the resulting ranks of biclusters generated when using different values of these weights. This exploratory analysis of the large set of obtained biclusters can be of key importance on domains where the user is not completely certain of what she/he is looking for, i.e. very open domains.

We applied the suggested methodology to our case study. Therefore, we defined a set of rows indexes (biological indexes) and set of columns indexes (geographical indexes). We also developed a web application, ORCA, which allows us to evaluate each of the generated biclusters using our methodology. ORCA also allows us to interactively define the weight of the biological score, the weight of the geographical score, and the weights for each index.

We used ORCA to compute the scores for each bicluster and to analyse the ones with the highest scores. Several interesting biclusters were found. Two of them unveiled some interesting relationships, which were unknown so far, between key microbial functions (nitrogen biogeochemistry) within different marine ecosystems.

Future Research Directions

Regarding the work presented in this dissertation we would like to propose some future research directions.

In our methodology, we assume that the maximum value of an index always corresponds to its optimal value. However, this is not always the case. For instance, the minimum value of the *Dispersion* index is in fact its optimal value. Thus, we normalized the values of the *Dispersion* index and then we computed the difference between 1 and the standard normalized values. In our case study, a low *Dispersion* value is always preferable. However, there are other indexes, such as the *nSamples* index, where the definition of what is its optimal value can change according to the goal of the analysis at hand. Hence, we can include an option in ORCA to allow

the users to choose what corresponds the optimal value of a certain index, i.e., if it corresponds to the maximum value or the minimum value of the index.

We can also include in `ORCA` more interactive plots using the package `plotly` [75]. Furthermore, we can also apply our methodology in other analysis, such as customer segmentation in marketing data.

On a more theoretical ground, we could explore whether it is feasible to incorporate our proposed evaluation criteria within the biclustering algorithms, i.e. try to bias the search for biclusters towards the optimisation of the user-defined evaluation criteria.

Appendix A

Experimental setup

All analysis were performed using `R` (version 3.2.4). The biclusters were generated with the `biclust` R package (version 1.2.0) [41]. Table A.1 presents the algorithms and the parameters used to generate our collection of biclusters.

`ORCA` was built using the packages `shiny` [17], `shinyjs` [6] and `DT` [94].

The plots for all the analysis were generated using the packages: `ggplot2` [89], `arulesViz` [31], `gplots` [86], `corrplot` [87], `ComplexHeatmap` [30], `GGally` [72], `ggmap` [40] and `rworldmap` [77].

The package `arules` [32] was used to generate association rules.

For data manipulation we used the packages: `readxl` [91], `dplyr` [93], `tidyr` [92], `lazyeval` [90], `reshape2` [88] and `Hmisc` [38].

In order to obtain geographical informations about the location from where each sample was taken, as the country and the continent, we used the package `RgoogleMaps` [52]. We computed the great circle distance between the sampling sites using the `WGS84` ellipsoid with the package `sp` [11, 63].

Algorithm	Arguments	Value
BCCC	delta	$[1.0, 1.1, 1.2, \ldots, 10.0]$
	alpha	1.5
	number	100
BCPlaid	cluster	b
	background	T
	shuffle	3
	row.release	$[0.6, 0.7]$
	col.release	$[0.6, 0.7]$
	max.layers	$[10, 20, 30]$
	iter.startup	$[5, 6, 7]$
	iter.layer	$[10, 11, 12]$
	back.fit	$[0, 1]$
	verbose	F
BCSpectral	normalization	log
	numberOfEigenvalues	3
	minr	$[2, 3]$
	minc	$[2, 3]$
	withinVar	$[1, 2]$
BCQuest	ns	10
	nd	$[10, 11, 12]$
	sd	$[5, 6, 7]$
	alpha	$[0.05, 0.06]$
	number	10

Table A.1: Experimental parameters.

Appendix B

Biclustering Analysis

As we mentioned in the section 5.3, we used ORCA to evaluate our collection of biclusters. Since several interesting biclusters were obtained, we performed several downstream analysis in order to understand the relations between the IPRs and the relations between the IPRs and the environmental variables. In this chapter we present the analysis that supported the findings presented in the section 5.3 by using ORCA.

B.1 Bicluster 6559

The bicluster with the index 6559 contains two IPRs (IPR030655 and IPR000392) involved in the N-fixation. We can observe from Figure B.1 the variation of the expression levels of each IPR along the samples (belonging to this bicluster). Figure B.2 illustrates the geographical location from where each sample belonging to this bicluster was taken. We can observe that most of the samples in this bicluster were taken in the European continent and are on the same latitudinal band. This led us to ask if there was some interesting environmental relation between the samples. Figure B.3 depicts the distribution of the environmental variables, after standardization, from the samples belonging to this bicluster. Table B.1 depicts the summarization for those environmental variables . The correlations between the environmental variables and

the IPRs were useful to the biological interpretation of the bicluster. Figure B.4 shows those significant correlations ($p\text{-}value = 0.05$). Oh the other hand, Figure B.5 shows all of those correlations, not just the significant ones. We also studied the significant correlations only between IPRs, depicted in Figure B.6, since that was useful to understand how the IPRs relate to each other.

	DepthWater	Temp	Salt
Min	0.00	11.90	31.00
1st Qu.	0.20	14.79	34.30
Median	1.00	18.09	34.59
Mean	3.90	18.99	34.94
3rd Qu.	2.00	23.43	35.14
Max	39.00	29.00	39.90

Table B.1: Summarization of environmental variables.

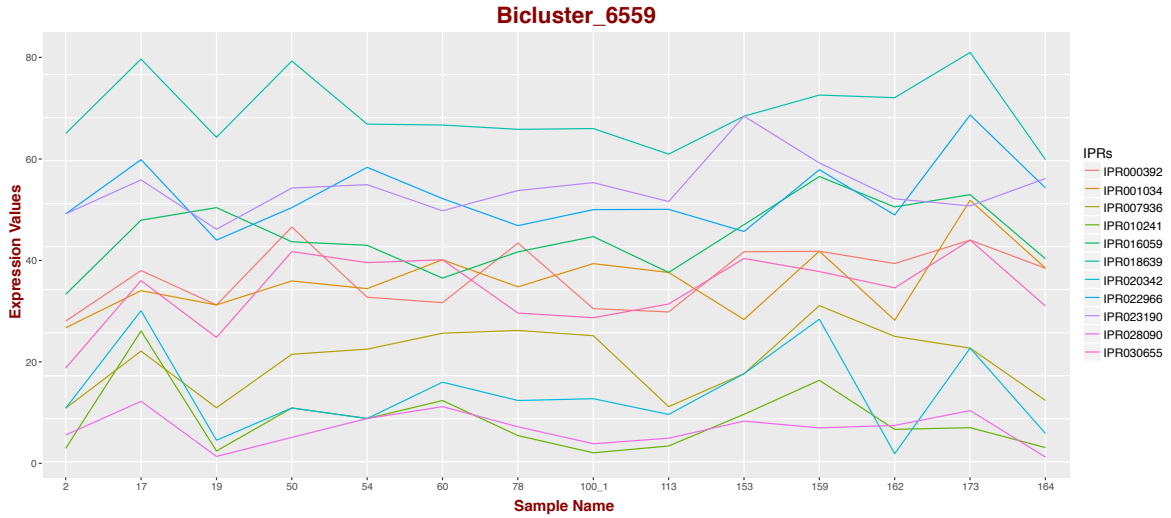


Figure B.1: Expression values of the IPRs along the samples.

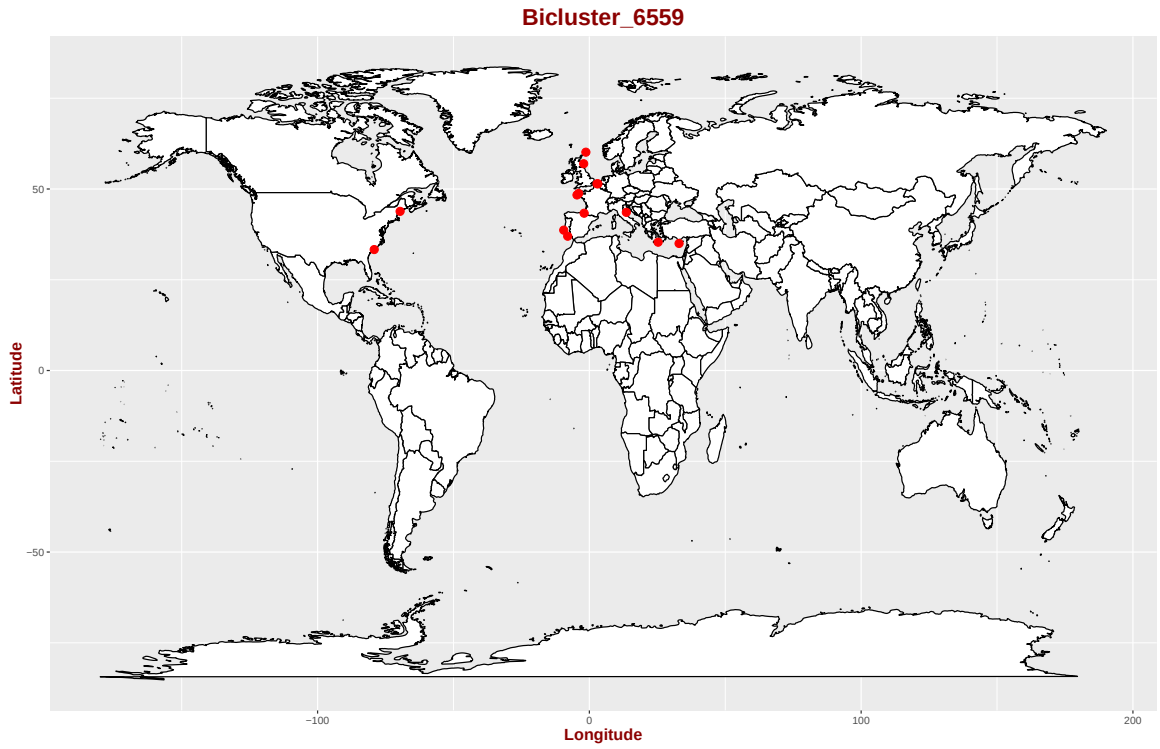


Figure B.2: Geographical distribution of the sampling sites.

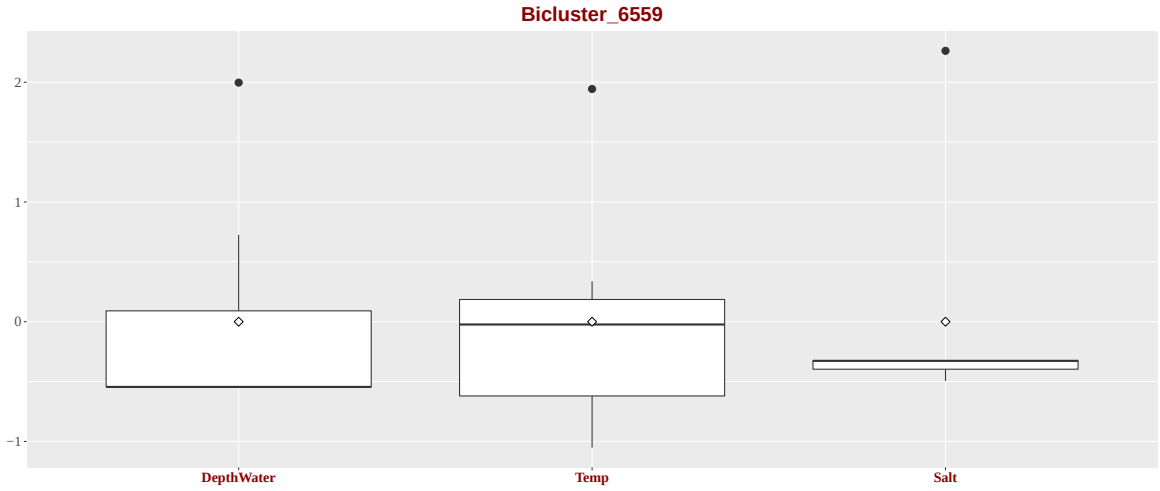


Figure B.3: Distribution of the environmental variables.

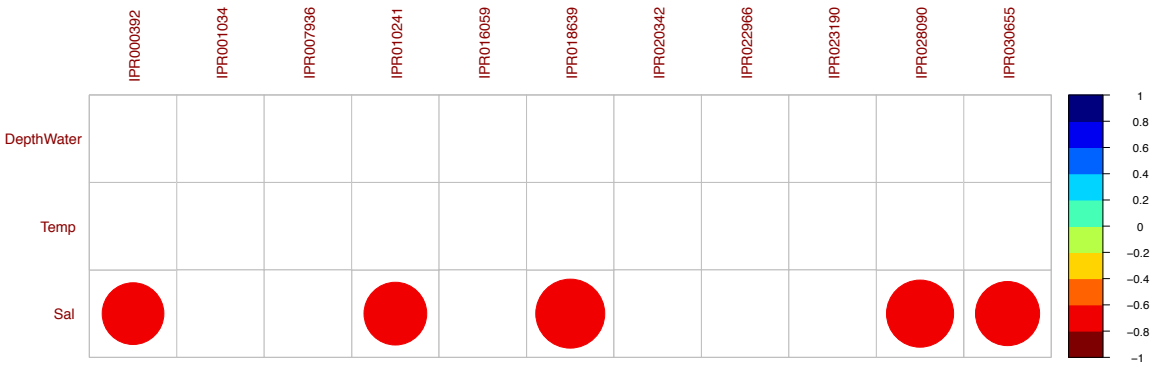


Figure B.4: Significant correlations between IPRs and the environmental variables.

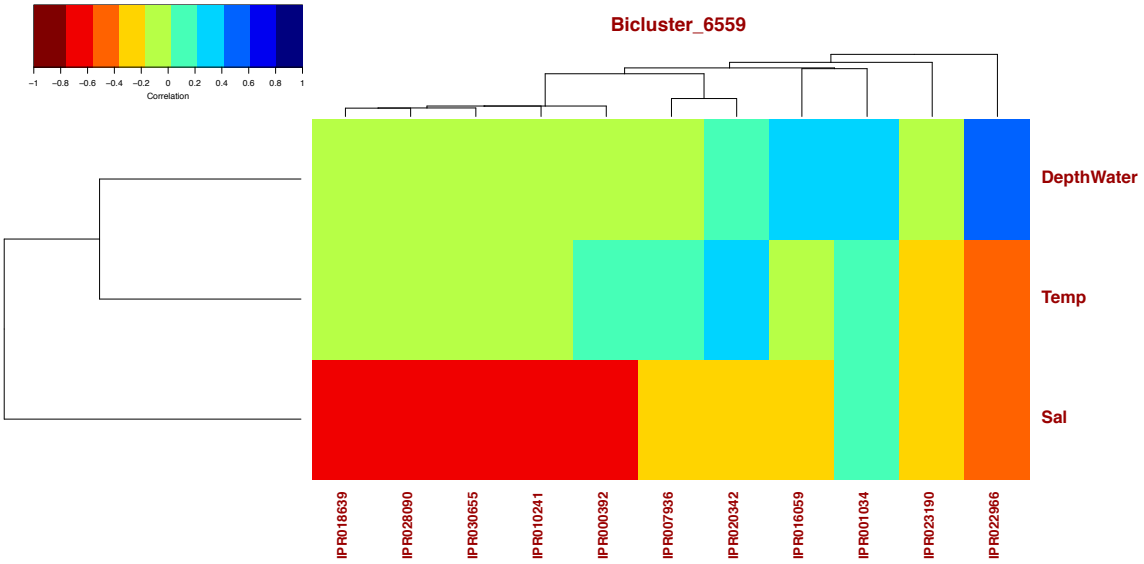


Figure B.5: Correlations between IPRs and the environmental variables.

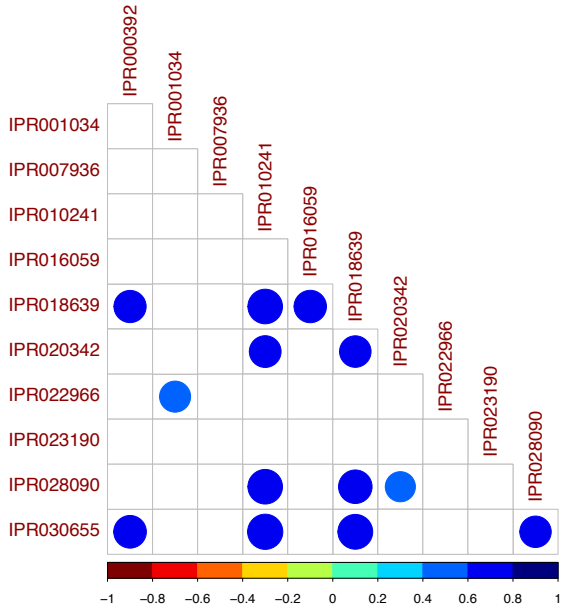


Figure B.6: Significant correlations between IPRs.

B.2 Bicluster 9093

The bicluster with the index 9093 contains one IPR (IPR006067) involved in the DNRA. In order to support the biological interpretation of this bicluster we did a similar analysis as the previous one. Figure B.7 shows the variation of the expression levels of each IPR along the samples, belonging to this bicluster. Figure B.8 illustrates the geographical location from where each sample belonging to this bicluster was taken. Most of the sampling sites are on the same longitudinal band. Once again, this led us to ask if there was some interesting environmental relation between the samples. Therefore, Figure B.9 depicts the distribution of the environmental variables, after standardization, from the samples belonging to this bicluster. Table B.2 depicts the summarization for those environmental variables. We also analysed several correlations. Figure B.10 shows the significant correlations between the environmental variables and the IPRs ($p\text{-}value = 0.05$). Oh the other hand, Figure B.11 shows all of those correlations, not just the significant ones. Finally, the significant correlations between IPRs are depicted in Figure B.12.

	DepthWater	Temp	Salt
Min	0.00	12.20	31.00
1st Qu.	0.00	16.00	34.30
Median	0.00	17.00	35.14
Mean	6.01	17.94	42.59
3rd Qu.	2.00	20.45	37.97
Max	50.00	23.60	100.00

Table B.2: Summarization of environmental variables.

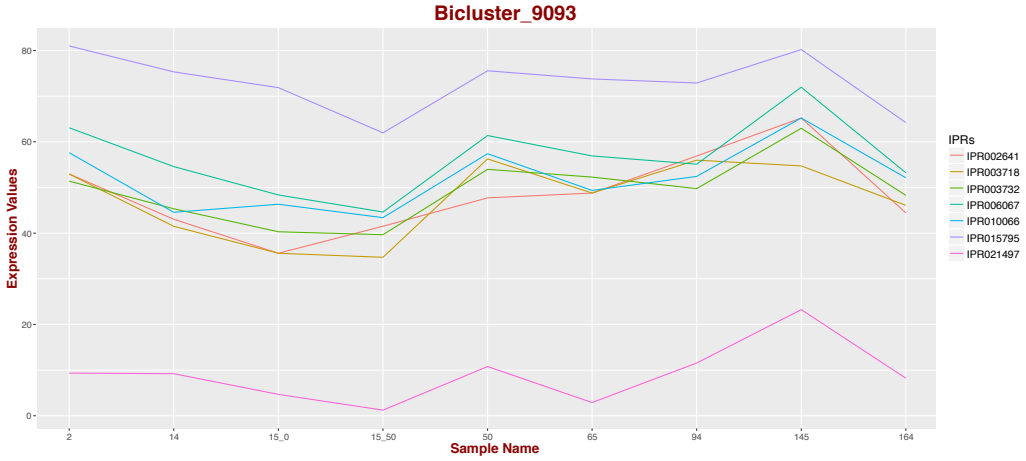


Figure B.7: Expression values of the IPRs along the samples.

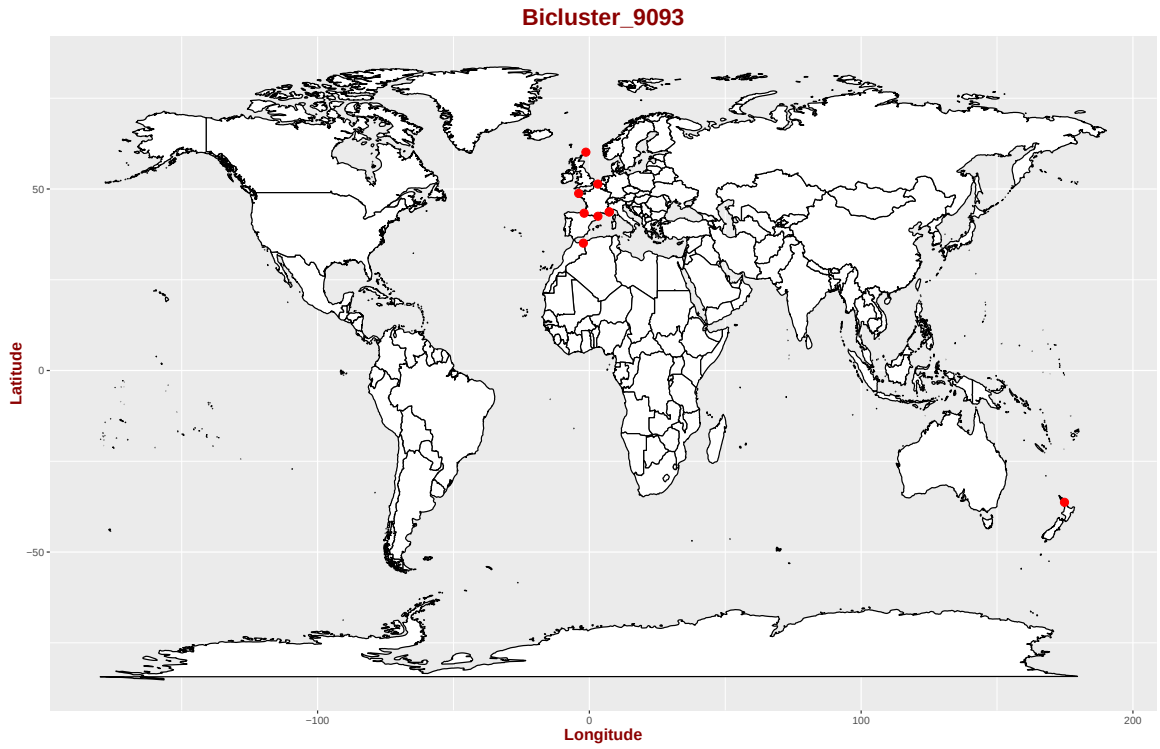


Figure B.8: Geographical distribution of the sampling sites.

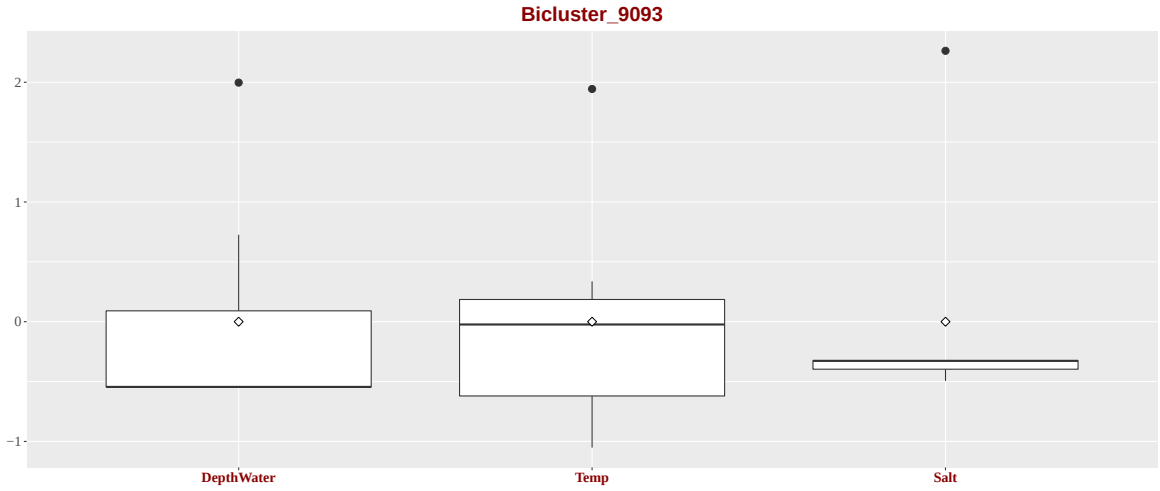


Figure B.9: Distribution of the environmental variables.

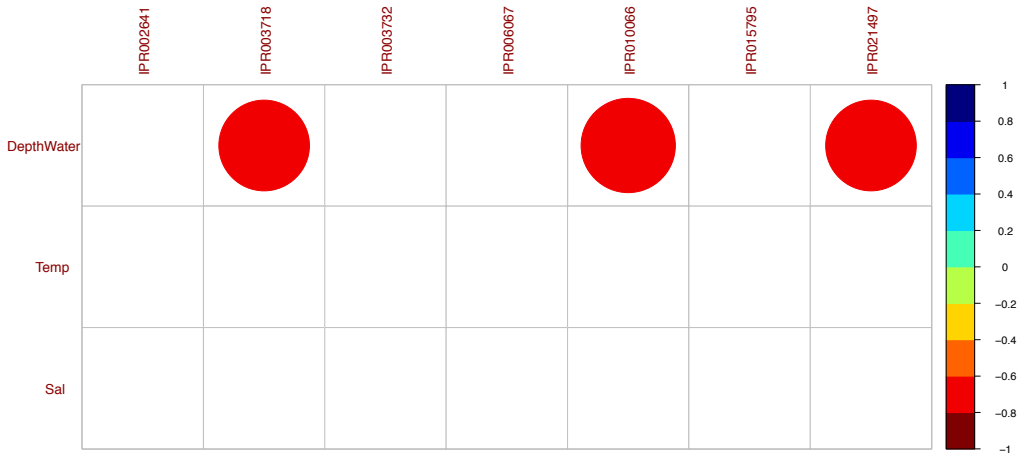


Figure B.10: Significant correlations between IPRs and the environmental variables.

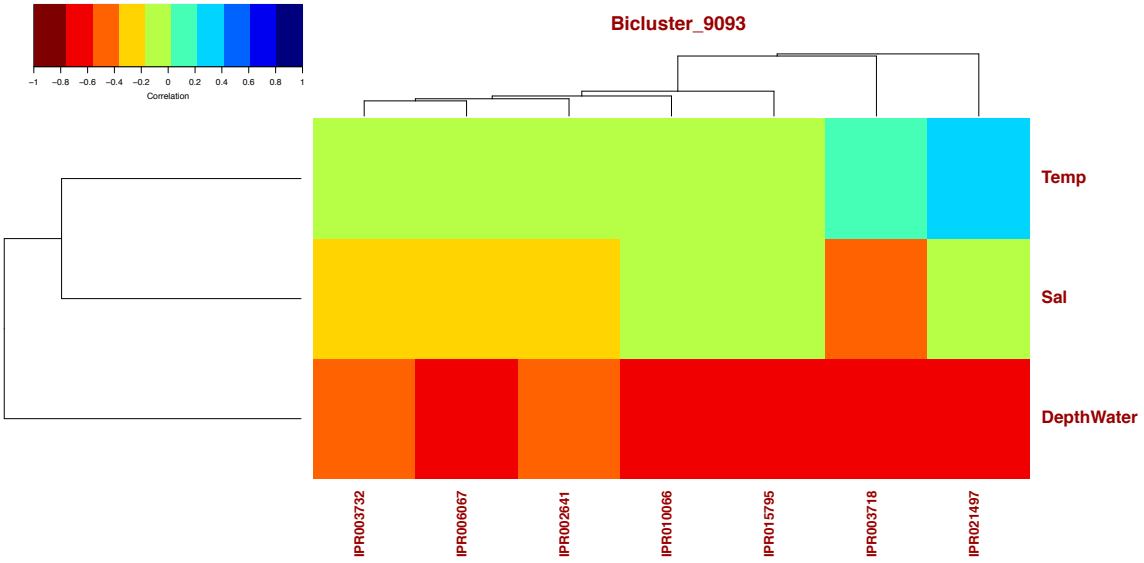


Figure B.11: Correlations between IPRs and the environmental variables.

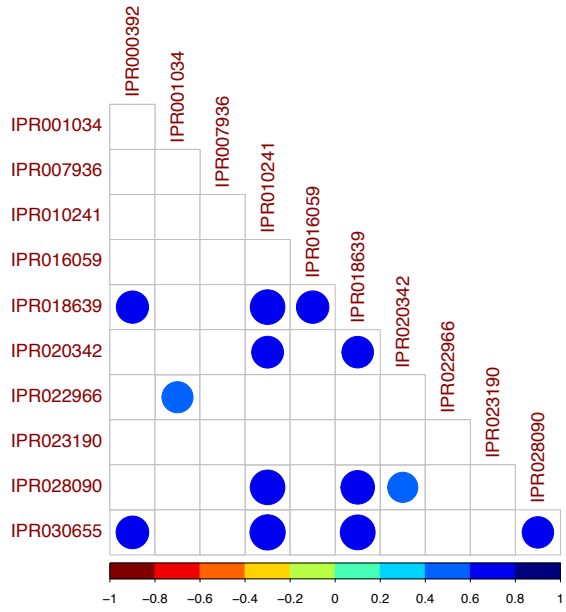


Figure B.12: Significant correlations between IPRs.

Bibliography

[1] C. C. Aggarwal. *Data Mining: The Textbook*. Springer Publishing Company, Incorporated, 2015.

[2] C. C. Aggarwal and C. K. Reddy. *Data Clustering: Algorithms and Applications*. Chapman & Hall/CRC, 1st edition, 2013.

[3] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, SIGMOD '98, pages 94–105, New York, NY, USA, 1998. ACM.

[4] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, June 1993.

[5] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: Tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.

[6] D. Attali. *shinyjs: Perform Common JavaScript Operations in Shiny Apps using Plain R Code*, 2016. R package version 0.6.

[7] P. Baldi and G. W. Hatfield. *DNA Microarrays and Gene Expression*. Cambridge University Press, 2002. Cambridge Books Online.

[8] R. E. Bellman. *Adaptive control processes - A guided tour*. Princeton University Press, 1961.

[9] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini. Discovering local structure in gene expression data: The order-preserving submatrix problem. In *Proceedings of the Sixth Annual International Conference on Computational Biology*, RECOMB '02, pages 49–57, New York, NY, USA, 2002. ACM.

[10] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of computational biology*, 6(3-4):281–297, 1999.

[11] R. S. Bivand, E. Pebesma, and V. Gomez-Rubio. *Applied spatial data analysis with R, Second edition*. Springer, NY, 2013.

[12] R. E. Bonner. On some clustering techniques. *IBM J. Res. Dev.*, 8(1):22–32, January 1964.

[13] C. Borgelt. Efficient implementations of apriori and eclat. In *Proc. 1st IEEE ICDM Workshop on Frequent Item Set Mining Implementations (FIMI 2003, Melbourne, FL). CEUR Workshop Proceedings 90*, page 90, 2003.

[14] S. Busygin, G. Jacobsen, E. Krämer, and C. Ag. Double conjugated clustering applied to leukemia microarray data. In *In 2nd SIAM ICDM, Workshop on clustering high dimensional data*, 2002.

[15] A. Califano, G. Stolovitzky, and Y. Tu. Analysis of gene expression microarrays for phenotype classification. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 75–85. AAAI Press, 2000.

[16] T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-Simulation and Computation*, 3(1):1–27, 1974.

[17] W. Chang, J. Cheng, J. J. Allaire, Y. Xie, and J. McPherson. *shiny: Web Application Framework for R*, 2016. R package version 0.13.1.

[18] Y. Cheng and G. M. Church. Biclustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 93–103. AAAI Press, 2000.

[19] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1(2):224–227, February 1979.

[20] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000.

[21] J. C. Dunn. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, 3(3):32–57, January 1973.

[22] K. Eren, M. Deveci, O. Küçüktunç, and Ü. V. Çatalyürek. A comparative analysis of biclustering algorithms for gene expression data. *Briefings in Bioinformatics*, 2012.

[23] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[24] V. Estivill-Castro. Why so many clustering algorithms: A position paper. *SIGKDD Explor. Newsl.*, 4(1):65–75, June 2002.

[25] C. Fraley and A. E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41:578–588, 1998.

[26] C. A. Francis, J. M. Beman, and M. M. Kuypers. New processes and players in the nitrogen cycle: the microbial ecology of anaerobic and archaeal ammonia oxidation. *The ISME Journal*, 1(5):19–27, 2007-05-01 00:00:00.001.

[27] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. *Proceedings of the National Academy of Sciences*, 97(22):12079–12084, 2000.

[28] J. A. Gilbert and C. L. Dupont. Microbial metagenomics: Beyond the genome. *Annual Review of Marine Science*, 3(1):347–371, 2011. PMID: 21329209.

[29] J. A. Gilbert, Janet K. Jansson, and Rob Knight. The earth microbiome project: successes and aspirations. *BMC Biology*, 12(1):1–4, 2014.

[30] Z. Gu. *ComplexHeatmap: Making Complex Heatmaps*, 2015. R package version 1.6.0.

[31] M. Hahsler and S. Chelluboina. *arulesViz: Visualizing Association Rules and Frequent Itemsets*, 2016. R package version 1.1-1.

[32] M. Hahsler, B. Gruen, and K. Hornik. arules – A computational environment for mining association rules and frequent item sets. *Journal of Statistical Software*, 14(15):1–25, October 2005.

[33] J. Han. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[34] J. Handl, J. Knowles, and D. B. Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15):3201–3212, August 2005.

[35] J. A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.

[36] P. Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, February 1912.

[37] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[38] F. E. Harrell Jr, with contributions from C. Dupont, and many others. *Hmisc: Harrell Miscellaneous*, 2016. R package version 3.17-2.

[39] J. H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

[40] D. Kahle and H. Wickham. ggmap: Spatial visualization with ggplot2. *The R Journal*, 5(1):144–161, 2013.

[41] S. Kaiser, R. Santamaria, T. Khamiakova, M. Sill, R. Theron, L. Quintales, F. Leisch, and E. De Troyer. *biclust: BiCluster Algorithms*, 2015. R package version 1.2.0.

[42] M. Kanehisa and S. Goto. Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30, 2000.

[43] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein. Spectral biclustering of microarray cancer data: Co-clustering genes and conditions. *Genome Research*, 13:703–716, 2003.

[44] A. Kopf, M. Bicak, R. Kottmann, J. Schnetzer, I. Kostadinov, K. Lehmann, A. Fernandez-Guerra, C. Jeanthon, E. Rahav, M. Ullrich, E. Sonnenschein, and S. Jackson. The ocean sampling day consortium. *GigaScience*, 4(27), 2015.

[45] H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans. Knowl. Discov. Data*, 3(1):1:1–1:58, March 2009.

[46] H.P. Kriegel, P. Kröger, J. Sander, and A. Zimek. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240, 2011.

[47] L. Lazzeroni and A. Owen. Plaid models for gene expression data. *Statistica Sinica*, 12:61–86, 2000.

[48] Y.-R. Lee, J.-H. Lee, and C.-H. Jun. Validation measures of bicluster solutions. *Industrial Engineering and Management Systems*, 8(2):101–108, 2009.

[49] J. Liu and W. Wang. Op-cluster: Clustering by tendency in high dimensional space. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM '03, pages 187–, Washington, DC, USA, 2003. IEEE Computer Society.

[50] X. Liu and L. Wang. Computing the maximum similarity bi-clusters of gene expression data. *Bioinformatics*, 23(1):50–56, 2007.

[51] S. P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28:129–137, 1982.

[52] M. Loecher and K. Ropkins. RgoogleMaps and loa: Unleashing R graphics power on map tiles. *Journal of Statistical Software*, 63(4):1–18, 2015.

[53] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 1(1):24–45, January 2004.

[54] C. Magalhães, J. Séneca, C. Leite, M. Monteiro, C. Bartilotti, A. do Santos, T. Kahlke, J. Tolman, R. Pires, R. Costa, and L. Torgo. Distribution and environmental controls on marine nitrogen biogeochemical functions. Paper presented at the 41st, CIESM, Kiel, 2016.

[55] M. Mahajan, P. Nimbhorkar, and K. Varadarajan. The planar k-means problem is np-hard. In *Proceedings of the 3rd International Workshop on Algorithms and Computation*, WALCOM '09, pages 274–285, Berlin, Heidelberg, 2009. Springer-Verlag.

[56] O. Maimon and L. Rokach. *Data Mining and Knowledge Discovery Handbook*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[57] B. Mirkin. *Mathematical Classification and Clustering.* Kluwer Academic Publishers, 1996.

[58] N. Mishra, R. Schreiber, I. Stanton, and R. E. Tarjan. *Clustering Social Networks*, pages 56–67. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[59] A. Mitchell, F. Bucchini, G. Cochrane, H. Denise, P. ten Hoopen, M. Fraser, S. Pesseat, S. Potter, M. Scheremetjew, P. Sterk, and R. D. Finn. Ebi metagenomics in 2016 - an expanding and evolving resource for the analysis and archiving of metagenomic data. *Nucleic Acids Research*, 2015.

[60] T. M. Murali and S. Kasif. Extracting conserved gene expression motifs from gene expression data. In *Pac. Symp. Biocomput*, pages 77–88, 2003.

[61] A. Oghabian, S. Kilpinen, S. Hautaniemi, and E. Czeizler. Biclustering methods: Biological relevance and application in gene expression analysis. *PLoS ONE*, 9(3):1–10, 03 2014.

[62] D. A. Pearce, I. A. Alekhina, A. Terauds, A. Wilmotte, A. Quesada, A. Edwards, A. Dommergue, B. Sattler, B. J. Adams, C. Magalhães, W.-L. Chu, M. C. Y. Lau, C. Cary, D. J. Smith, D. H. Wall, G. Eguren, G. Matcher, J. A. Bradley, J.-P. de Vera, J. Elster, K. A. Hughes, L. Cuthbertson, L. G. Benning, N. Gunde-Cimerman, P. Convey, S. G. Hong, S. B. Pointing, V. H. Pellizari, and W. F. Vincent. Aerobiology over antarctica – a new initiative for atmospheric ecology. *Frontiers in Microbiology*, 7:16, 2016.

[63] E. J. Pebesma and R. S. Bivand. Classes and methods for spatial data in R. *R News*, 5(2):9–13, November 2005.

[64] R. Peeters. The maximum edge biclique problem is np-complete. *Discrete Appl. Math.*, 131(3):651–654, September 2003.

[65] J. Pei, X. Zhang, M. Cho, H. Wang, and P. S. Yu. Maple: A fast algorithm for maximal pattern-based clustering. In *Proceedings of the Third IEEE International*

Conference on Data Mining, ICDM '03, pages 259–, Washington, DC, USA, 2003. IEEE Computer Society.

[66] B. Pontes, R. Giráldez, and J. S. Aguilar-Ruiz. Biclustering on expression data. *J. of Biomedical Informatics*, 57(C):163–180, October 2015.

[67] A. Prelić, S. Bleuler, P. Zimmermann, A. Wille, P. Bühlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22(9):1122–1129, 2006.

[68] G. Punj and D. W Stewart. Cluster analysis in marketing research: Review and suggestions for application. *Journal of marketing research*, pages 134–148, 1983.

[69] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[70] P. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, November 1987.

[71] R. Santamaría, L. Quintales, and R. Therón. Methods to bicluster validation and comparison in microarray data. In *Proceedings of the 8th International Conference on Intelligent Data Engineering and Automated Learning*, IDEAL'07, pages 780–789, Berlin, Heidelberg, 2007. Springer-Verlag.

[72] B. Schloerke, J. Crowley, D. Cook, F. Briatte, M. Marbach, E. Thoen, and A. Elberg. *GGally: Extension to ggplot2*, 2016. R package version 1.0.1.

[73] S. Z. Selim and M. A. Ismail. K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(1):81–87, January 1984.

[74] J. R. Seymour. A sea of microbes: the diversity and activity of marine microorganisms. *Microbiology Australia*, 35(4), 2014.

[75] C. Sievert, C. Parmer, T. Hocking, S. Chamberlain, K. Ram, M. Corvellec, and P. Despouy. *plotly: Create Interactive Web Graphics via 'plotly.js'*, 2016. R package version 4.5.2.

[76] R. R. Sokal and F. J. Rohlf. The comparison of dendrograms by objective methods. *Taxon*, 11(2):33–40, 1962.

[77] A. South. rworldmap: A new r package for mapping global data. *The R Journal*, 3(1):35–43, June 2011.

[78] A. Strehl, E. Strehl, J. Ghosh, and R. Mooney. Impact of similarity measures on web-page clustering. In *In Workshop on Artificial Intelligence for Web Search (AAAI 2000*, pages 58–64. AAAI, 2000.

[79] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[80] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18(suppl 1):S136–S144, 2002.

[81] A. Tanay, R. Sharan, and R. Shamir. Biclustering Algorithms. In *Handbook of Computational Molecular Biology*, Chapman & Hall/CRC Computer & Information Science Series, pages 17–26. Chapman and Hall/CRC, dec 2005.

[82] C. Tang, L. Zhang, M. Ramanathan, and A. Zhang. Interrelated two-way clustering: An unsupervised approach for gene expression data analysis. In *Proceedings of the 2Nd IEEE International Symposium on Bioinformatics and Bioengineering*, BIBE '01, pages 41–, Washington, DC, USA, 2001. IEEE Computer Society.

[83] H. Turner, T. Bailey, and W. Krzanowski. Improved biclustering of microarray data demonstrated through systematic performance tests. *Computational Statistics and Data Analysis*, 48(2):235 – 254, 2005.

[84] H. Wang, W. Wang, J. Yang, and P. S. Yu. Clustering by pattern similarity in large data sets. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, SIGMOD '02, pages 394–405, New York, NY, USA, 2002. ACM.

[85] W. Wang, J. Yang, and R. R. Muntz. Sting: A statistical information grid approach to spatial data mining. In *Proceedings of the 23rd International Conference on Very Large Data Bases*, VLDB '97, pages 186–195, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[86] G. R. Warnes, B. Bolker, L. Bonebakker, R. Gentleman, W. H. A. Liaw, T. Lumley, M. Maechler, A. Magnusson, S. Moeller, M. Schwartz, and B. Venables. *gplots: Various R Programming Tools for Plotting Data*, 2015. R package version 2.17.0.

[87] T. Wei. *corrplot: Visualization of a correlation matrix*, 2013. R package version 0.73.

[88] H. Wickham. Reshaping data with the reshape package. *Journal of Statistical Software*, 21(12):1–20, 2007.

[89] H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2009.

[90] H. Wickham. *lazyeval: Lazy (Non-Standard) Evaluation*, 2015. R package version 0.1.10.

[91] H. Wickham. *readxl: Read Excel Files*, 2015. R package version 0.1.0.

[92] H. Wickham. *tidyr: Easily Tidy Data with 'spread()' and 'gather()' Functions*, 2016. R package version 0.4.1.

[93] H. Wickham and R. Francois. *dplyr: A Grammar of Data Manipulation*, 2015. R package version 0.4.3.

[94] Y. Xie. *DT: A Wrapper of the JavaScript Library 'DataTables'*, 2015. R package version 0.1.56.

[95] J. Yang, H. Wang, W. Wang, P. Yu, U. Ibm, U. Chapel, H. Ibm, T. J. Watson, and T. J. Watson. Enhanced biclustering on expression data. In *Proc. of 3rd IEEE Symposium on BioInformatics and BioEngineering (BIBE'03*, pages 321–327, 2003.

[96] J. Yang, W. Wang, H. Wang, and P. Yu. delta-clusters: Capturing subspace correlation in a large data set. In *Proc. of 18th IEEE Intern. Conf. on Data Engineering*, 2002.

[97] Seung Yon Rhee, V. Wood, K. Dolinski, and S. Draghici. Use and misuse of the gene ontology annotations. *Nature Reviews Genetics*, 9(7):509–515, 7 2008.

[98] H. Zhang and K. Ning. The tara oceans project: New opportunities and greater challenges ahead. *Genomics, Proteomics and Bioinformatics*, 13(5):275 – 277, 2015. SI: Metagenomics of Marine Environments.